AD NUMBER
ADB007943
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; AUG 1975. Other requests shall be referred to Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH 45433.**
AUTHORITY
AFFDL ltr 2 May 1979

AFFDL-TR-75-103

(2)

COMPUTER PROGRAMS FOR CALCULATING SMALL DISTURBANCE TRANSONIC FLOWS ABOUT OSCILLATING PLANAR WINGS

SCIENCE APPLICATIONS, INCORPORATED

AUGUST 1975

TECHNICAL REPORT AFFDL-TR-75-103
REPORT FOR PERIOD JULY 1974 – AUGUST 1975

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

William L. Holman
WILLIAM L. HOLMAN, 1Lt, USAF
Project Engineer
Design Analysis Group
Aerospace Dynamics Branch

FOR THE COMMANDER

James J. Olsen
JAMES J. OLSEN
Acting Assistant for Research & Technology
Vehicle Dynamics Division

Copies of this report should not be returned unless return is required by security considerations, contractural obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFFDL-TR-75-103	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): COMPUTER PROGRAMS FOR CALCULATING SMALL DISTURBANCE TRANSONIC FLOWS ABOUT OSCILLATING PLANAR WINGS.	5. TYPE OF REPORT & PERIOD COVERED: Users Manual, Nov [redacted]74 – Aug [redacted]75.
	6. PERFORMING ORG. REPORT NUMBER: SAI-76-547-LA
7. AUTHOR(s): J. L. Farr, Jr., R. M. Traci, E. D. Albano	8. CONTRACT OR GRANT NUMBER(s): F33615-74-C-3094
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Science Applications, Inc., 101 Continental Building, Suite 310, El Segundo, California 90245	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project 1370, [redacted] 137004
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Flight Dynamics Laboratory, Aerospace Dynamics Branch, FYS, Wright-Patterson AFB, Ohio 45433	12. REPORT DATE: Aug[redacted]75
	13. NUMBER OF PAGES: 99 107p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Air Force Flight Dynamics Laboratory (same as above)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution limited to U.S. Government agencies only; test and evaluation, statement applied August 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FY), Wright-Patterson AFB, Ohio 45433.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

N/A

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Programs
Relaxation Methods
Unsteady Aerodynamics
Transonic Flow
Oscillating Airfoils and Wings
Flutter
Small Disturbance Theory

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Computer programs are described which implement a small disturbance potential flow theory for the three-dimensional unsteady transonic flow about rectangular planar wings undergoing harmonic oscillations. The theory is based upon the treatment of the unsteady flow as a small perturbation to the steady transonic flow. Separating the perturbation potential into a steady and unsteady component results in a pair of coupled boundary value → to N73B

DD FORM 1 JAN 73 1473A EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DN 407842

from 1473A

problems for the components. The governing equation for the steady perturbation potential is the usual nonlinear transonic potential equation and it is solved in computer program TDSTRN using the mixed differencing relaxation procedure of Murman and Cole. The governing equation for the unsteady perturbation potential is linear and, for the harmonic boundary disturbance considered, of mixed elliptic hyperbolic type depending on the local nature of the steady potential. Using a steady solution previously generated by TDSTRN computer program TDUTRN solves the unsteady potential equation by the same relaxation procedure. The solution procedures are found to be quite efficient, permitting the calculation of unsteady aerodynamic forces to engineering accuracy in a few minutes on a CDC 6600 computer.

1473B

FOREWORD

This computer program User's Manual was prepared by the Los Angeles Division of Science Applications, Incorporated, for the Vehicle Dynamics Division of the Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio. The computer programs were developed under Project 1370, "Dynamic Problems in Flight Vehicles", Task 137004, "Design Analysis", Contract F33615-74-C-3094. James J. Olsen and later Lt. William L. Holman (AFFDL/FYS) were the Air Force Task Engineers.

R. M. Traci was the principal investigator for the study and J. L. Farr, Jr., developed the computer programs described in this report. Consultant E. D. Albano contributed to the development and implementation of the numerical method.

The authors submitted this report in July 1975 for publication as an AFFDL technical report.

Other reports prepared and submitted under the aforementioned contract are: AFFDL-TR-74-37, "Small Disturbance Transonic Flows about Oscillating Airfoils," AFFDL-TR-74-135, "Computer Programs for Calculating Small Disturbance Transonic Flows about Oscillating Airfoils," AFFDL-TR-75-100, "Small Disturbance Transonic Flows about Oscillating Airfoils and Planar Wings."

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

Computer programs TDSTRN and TDUTRN implement a small disturbance potential flow theory for three-dimensional unsteady transonic flow about thin planar wings undergoing harmonic oscillation. The theory is based on the fact that a linear system can be obtained by considering the unsteady flow as a small perturbation to the non-uniform mean flow. The perturbation expansion approach has recently been developed with different emphasis in independent studies by the present authors [1,2,3] and Ehlers[4]. Detailed descriptions of the theory and numerical solution method used in the three-dimensional version of the programs, documented in the users manual, are presented in References 1-3. The method is a generalization of that described for two-dimensional airfoils in Reference 1 and of the 2-D computer programs documented in Reference 2. The final report of the present phase of research[3] describes the generalization and presents some illustrative results.

In the perturbation expansion approach used, the perturbation potential function is expanded in a series of increasing powers of a small parameter which is a measure of the amplitude of an unsteady disturbance to the boundary. The resulting expansion of the unsteady potential equation results in a sequence of partial differential equations for the perturbation potentials. The zeroth order equation is the usual nonlinear steady transonic potential equation of mixed elliptic/hyperbolic type and is solved in TDSTRN using the mixed differencing, relaxation procedure of Murman and Cole[5]. The first order unsteady potential equation is linear and for harmonic boundary disturbances is also of the mixed elliptic/hyperbolic type, depending upon the steady solution. It is solved in TDUTRN using the same numerical technique as used in TDSTRN.

The theory and practice of the computer program operation are discussed in the following sections. The small perturbation theory and numerical solution procedure are summarized in Sections 2.0 and 3.0, respectively. A description of the program's logical operation and a brief subroutine description are given in Section 4.0. Section 5.0 presents a complete description of the program input, with suggested values for various control variables, and the program output. Section 6.0 describes the program usage and includes suggestions for making effective use of the programs. Sample cases which exercise all program options are presented in Section 7.0 with a complete specification of all input and sample output. Finally, complete FORTRAN listings of TDSTRN and TDUTRN are presented in the appendices.

2.0 SMALL PERTURBATION THEORY FOR UNSTEADY TRANSONIC FLOW

Small disturbance theory is the principal analytical tool for all speed ranges and has become increasingly important in the transonic speed range in recent years. The general theory including the unsteady small perturbation approach used in this work is summarized in this section. The required numerical solution methods for the steady and unsteady systems are described in Section 3.0. It is noted that, following usual programming practice, the FORTRAN variables used in TDSTRN and TDUTRN are descriptive of the physical variables.

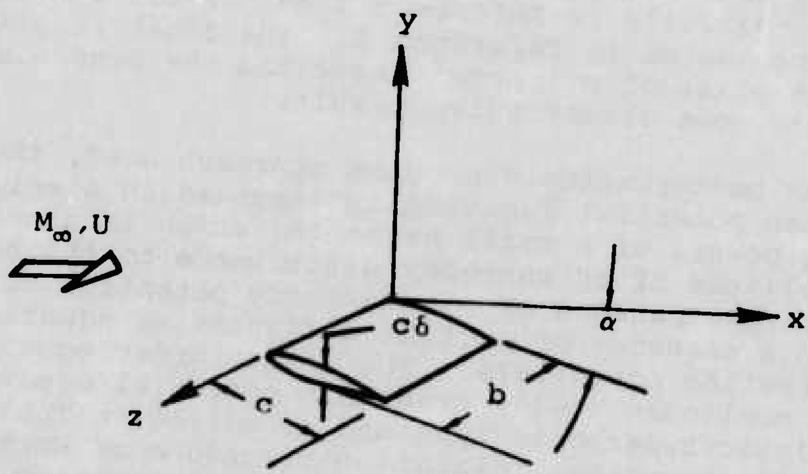

FIGURE 1. SCHEMATIC OF THREE-DIMENSIONAL PLANAR WING

The problem of interest is the flow about an airfoil (two-dimensional) or planar wing (three-dimensional) oscillating with various flexible or rigid body degrees of freedom in the transonic speed range. The airfoil geometry, flowfield schematic and coordinate definition are given in Figure 1 above. Rectangular coordinates (x,y,z) are fixed to the airfoil leading edge with origin at the wing root and U, M_∞, a_∞ are the freestream velocity, Mach number and sound speed respectively. The airfoil has a thickness ratio δ, which is the airfoil maximum thickness divided by its chord c, and angle of attack α and a semi-span b (ZSPAN). The assumption is made that δ<<1 and α is of the same order of magnitude as δ.

Also, the oscillatory motion of the airfoil is assumed to be described by a small non-dimensional displacement $\varepsilon << \delta$ and a reduced frequency $k = \omega c/U$ based on airfoil chord where ω is the frequency of oscillation.

Assuming inviscid, isentropic flow, the problem can be reduced to the solution of a single equation for a velocity potential plus the tangency boundary condition on the airfoil surface. As is well known, the derivation of a small disturbance theory for transonic flows requires a singular perturbation approach. The following scaling is thereby introduced:

$$\tilde{x} = \frac{x}{c},\ \tilde{y} = [(1+\gamma)\delta M_\infty^2]^{1/3}\,\frac{y}{c} \qquad \tilde{z} = [(1+\gamma)\delta M_\infty^2]^{1/3}\,\frac{z}{c}$$

$$\tilde{t} = \frac{[(1+\gamma)\delta M_\infty^2]^{2/3}}{M_\infty^2}\,\frac{U}{c}\,t \tag{1}$$

and the total potential is expanded about the uniform flow:

$$\psi = Uc\tilde{x} + \frac{\delta^{2/3}\,Uc}{[(1+\gamma)M_\infty^2]^{1/3}}\,\tilde{\phi}(\tilde{x},\tilde{y},\tilde{z},\tilde{t}) + \ldots \tag{2}$$

Retaining all terms of leading order in to total potential equation and boundary conditions results in the following form for the unsteady small disturbance system.

$$(K - \tilde{\phi}_{\tilde{x}})\tilde{\phi}_{\tilde{x}\tilde{x}} + \tilde{\phi}_{\tilde{y}\tilde{y}} + \tilde{\phi}_{\tilde{z}\tilde{z}} = 2\tilde{\phi}_{\tilde{x}\tilde{t}} + \underline{\frac{k}{\Omega}\,\tilde{\phi}_{\tilde{t}\tilde{t}}} \tag{3}$$

where the transonic similarity parameters are:

$$K = \frac{(1-M_\infty^2)}{[(1+\gamma)\delta M_\infty^2]^{2/3}} \qquad \Omega = \frac{M_\infty^2}{[(1+\gamma)\delta M_\infty^2]^{2/3}}\,k$$

with boundary conditions:

$$\tilde{\phi}_{\tilde{y}} = \left(\frac{\partial}{\partial\tilde{x}} + \underline{\frac{k}{\Omega}\frac{\partial}{\partial\tilde{t}}}\right) f_{u,\ell}(\tilde{x},\tilde{z},\tilde{t})$$

$$\text{on } y = \pm 0 \quad \begin{cases} 0 \le \tilde{x} \le 1 \\ 0 \le \tilde{z} \le b \end{cases} \tag{4}$$

$$\left[\tilde{\phi}_{\tilde{x}} + \underline{\frac{k}{\Omega}\tilde{\phi}_{\tilde{t}}}\right] = 0, \text{ on } \tilde{y}=0 \quad \begin{cases} \tilde{x} > 1 \\ 0 \le \tilde{z} \le b \end{cases} \tag{5}$$

$$\tilde{\phi}_{\tilde{x}}^2 + \tilde{\phi}_{\tilde{y}}^2 + \tilde{\phi}_{\tilde{z}}^2 \to 0 \text{ as } \tilde{x}^2+\tilde{y}^2+\tilde{z}^2 \to \infty \tag{6}$$

where $f_{u,\ell}$ is the unsteady airfoil shape function (Equation (7) below) on the upper and lower surfaces respectively, and where [] denotes a jump in the enclosed quantity between $y=0^-$ and 0^+. It is noted that the airfoil tangency boundary condition (Equation 4) and the Kutta condition (Equation 5) are applied in the small disturbance manner on y=0.

The system of Equations 3-6 provides a formulation of the unsteady airfoil problem in the non-linear domain, which includes flowfields with shocks. Certain terms in the above system are underlined as they may be omitted for a low frequency $[k \sim O(\delta^{2/3})]$ approximation. The present version of TDUTRN includes the low frequency approximation (IØPT=0) or general frequency formulation (IØPT=1) and either can be used at the discretion of the user.

The approach presented herein for solving the non-linear system given above (Equations 3-6) is to expand the perturbation potential function in terms of the unsteady boundary disturbance ε<<1. From this point on all tildas (∿) will be

dropped with the understanding that all variables are scaled variables. Harmonic boundary disturbances are explicitly treated:

$$f(x,z,t) = f_o(x,z) + \varepsilon f_\varepsilon(x,z)e^{i\Omega t} \tag{7}$$

and the perturbation potential is expanded as follows:

$$\phi(x,y,z,t) = \phi^0(x,y,z) + \varepsilon\, \phi^1(x,y,z)e^{i\Omega t} + \dots \tag{8}$$

Substituting this into the perturbation potential equation plus boundary conditions and combining terms results in the following pair of boundary value problems for ϕ^0 and ϕ^1 respectively. (In the following text, the superscript has been dropped from ϕ^1.)

$$\left.\begin{aligned}
&(K-\phi^0_x)\phi^0_{xx} + \phi^0_{yy} + \phi^0_{zz} = 0\\
&\phi^0_y = f'_o(x,z), \quad \text{on } y = \pm 0 \begin{cases} 0 \le x \le 1\\ 0 \le z \le b\end{cases}\\
&[\phi^0_x] = 0, \quad \text{on } y = 0 \begin{cases} x > 1\\ 0 < z < b\end{cases}\\
&(\phi^0_x)^2 + (\phi^0_y)^2 + (\phi^0_z)^2 \to 0 \quad \text{as } x^2+y^2+z^2 \to \infty
\end{aligned}\right\} \tag{9}$$

and

$$\left.\begin{aligned}
&(K-\phi^0_x)\phi_{xx} + \phi_{yy} + \phi_{zz} - (\phi^0_{xx} + 2i\Omega)\phi_x + \underline{k\Omega\phi} = 0\\
&\phi_y = f'_\varepsilon + \underline{ikf_\varepsilon}, \quad \text{on } y = \pm 0 \begin{cases} 0 \le x \le 1\\ 0 \le z \le b\end{cases}\\
&[\phi_x + \underline{ik\phi}] = 0, \quad \text{on } y = 0 \begin{cases} x > 1\\ 0 \le z \le b\end{cases}\\
&(\phi_x)^2 + (\phi_y)^2 + (\phi_z)^2 \to 0, \quad \text{as } x^2+y^2+z^2 \to \infty
\end{aligned}\right\} \tag{10}$$

System 9 is recognized as the usual formulation for steady transonic flow and system 10 is the formulation for the unsteady perturbation thereof. It is noted that the governing equation for ϕ is linear but of the same mixed elliptic/hyperbolic type as the steady solution. It is also noted that ϕ is in general complex thereby permitting phase shifts between field quantities and the boundary disturbance. As before, underlined terms in system 10 are neglected for a consistent low frequency approximation. Also for two dimensional airfoil sections, the z dependence on all quantities and the ϕ_{zz} terms in the equations are neglected.

The main physical quantities of interest are the pressure coefficient and airfoil force coefficients. The pressure coefficient, defined in the usual manner, is given by:

$$C_p = \frac{\delta^{2/3}}{[(1+\gamma)M_\infty^2]^{1/3}} (\bar{C}_p^0 + \epsilon\bar{C}_p e^{i\Omega t}) \qquad (11)$$

where the steady and unsteady scaled pressure coefficients are given to leading order in the small disturbance approximation by:

$$\bar{C}_p^0 = -2\phi_x^0, \quad \bar{C}_p = -2(\phi_x + \underline{ik\phi}) \qquad (12)$$

The formulations of the boundary value problems are essentially complete with the exception of the practical matter of setting the boundary conditions away from the airfoil, which depends on the particular problem; subsonic or supersonic free field, wind tunnel wall etc. Asymptotic far field solutions to Equations 10 have been developed for two-dimensional subsonic or supersonic free air or wind tunnel flows and for three-dimensional subsonic flow. These solutions are described in the present three-dimensional subsonic free air version of the computer programs.

3.0 NUMERICAL SOLUTION METHOD

The numerical solution procedure for the boundary value problems for the steady and unsteady perturbation potential, is based on the mixed differencing, line relaxation procedure developed by Murman, Cole and Krupp[5,6]. They pointed out the essential ingredient for the success of relaxation procedures for the steady transonic potential equation. The key to the approach is to account for the local nature of the flow (elliptic in subsonic regions, hyperbolic is supersonic regions) in the finite difference approximation to the governing equations. The solution method used in the present work for the steady perturbation potential, ϕ^0, is patterned after the method for general lifting airfoils developed by Krupp[6].

The application of the theory and solution method to two-dimensional airfoil sections presented in previous work are interesting and illustrative but for practical application to dynamics or flutter problems three-dimensional effects must be considered. As with most other effects, 3-D effects are more important at transonic speeds than in the other speed ranges. The efficiency of the present scheme is such that realistic three-dimensional computations are practical on modern computers and it is the purpose of this section to describe the generalization of the numerical solution procedure to permit 3-D calculations (Section 3.1).

The initial development of the method is restricted to rectangular planforms undergoing oscillations symmetric with respect to the wing root ($z = 0$). The small disturbance analysis and the unsteady perturbation theory valid for three-dimensional flows were described in Section 2. As indicated there, the generalization to three dimensions requires but the addition of the ϕ_{zz} term to the governing equations for the steady and unsteady perturbation potentials. Asymptotic solutions to the governing equations have been derived for lifting wings in subsonic free-stream flow by Klunker[7], for the steady flow, and by the present authors for the unsteady perturbation. These solutions are summarized in Section 3.2 and used in the numerical solution method to fix farfield boundary conditions. Three dimensional solutions for steady transonic flow have been presented by Bailey and Steger[8] and Newman and Klunker[9]; the latter work being most closely related to the method for steady flows used in this work. Extensions of the solution method for the unsteady perturbation parallel the steady method and these are now described.

3.1 Finite Difference Solution Method

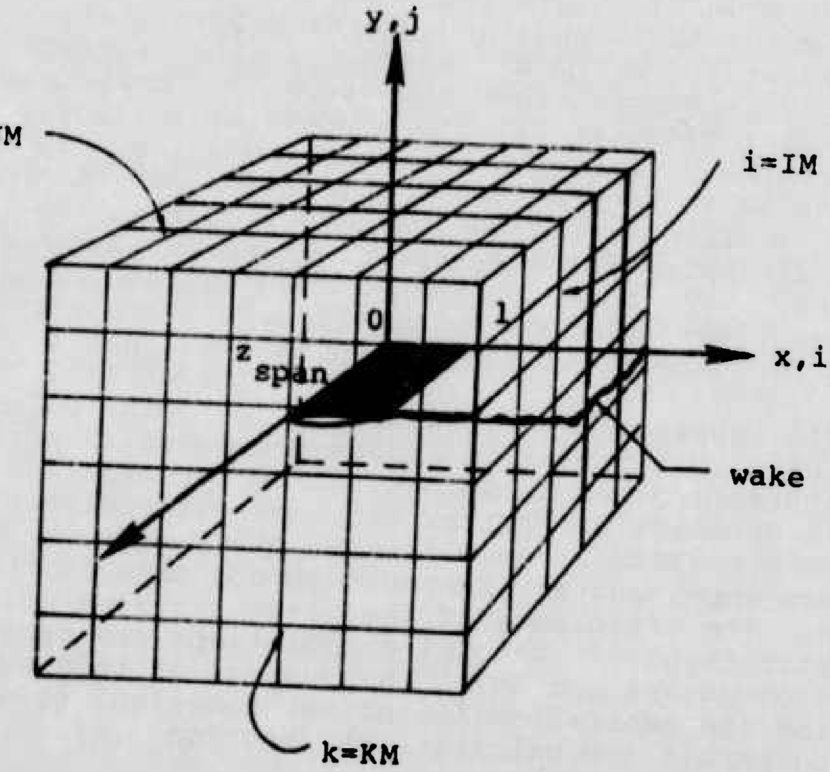

FIGURE 2. SCHEMATIC OF NUMERICAL SOLUTION DOMAIN

The three-dimensional numerical scheme constitutes the most straightforward extension of the two-dimensional method previously described in detail in Reference 1. As shown schematically in Figure 2, a cubic rectangular mesh of finite extent with uneven grid line spacing is overlayed on the 3-D solution space. The grid is concentrated near the airfoil and expanded out to the far boundaries of the grid. The finite difference equations are identical to the corresponding two-dimensional versions[1,2] with the addition of a centered difference form for ϕ_{zz} given by:

$$\phi_{zz_{i,j,k}} = \frac{2}{(\Delta z_k + \Delta z_{k-1})} \left\{ \frac{1}{\Delta z_k} (\phi_{i,j,k+1} - \phi_{i,j,k}) - \frac{1}{\Delta z_{k-1}} (\phi_{i,j,k} - \phi_{i,j,k-1}) \right\} \tag{13}$$

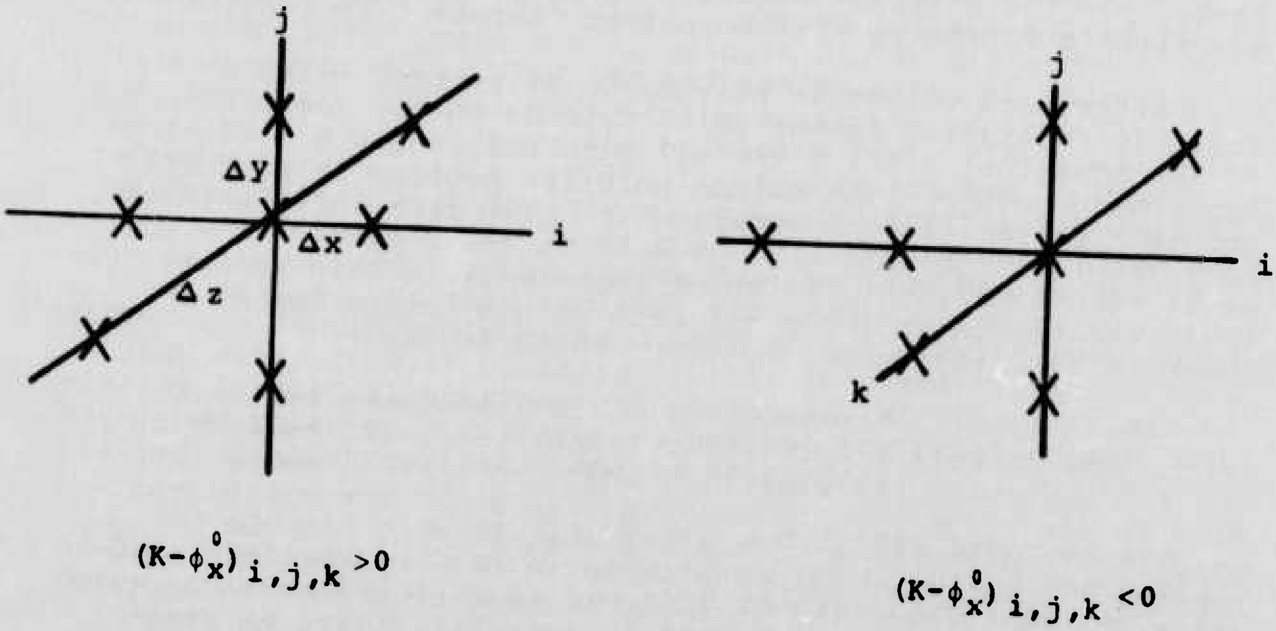

FIGURE 3. SCHEMATIC OF DIFFERENCE SCHEME

The computational star for the three-dimensional scheme is shown schematically in Figure 3 above. The tests for the elliptic on hyperbolic nature of the equation are made on the centered difference form of $(K-\phi^0_x)_{i,j,k}$, and depending on the value of this coefficient the x derivatives of ϕ are centered or backward differenced as in the 2-D case. A parabolic point operator is used in both the steady and unsteady schemes and a shock point operator is used in the finite difference scheme for the steady potential.

As before, the finite difference equations are set up for each column (x,z = constants) in turn, taking account of the airfoil, wake and farfield boundary conditions. In the steady solution this results in a set of quadratic equations for the column of ϕ's which are solved by linearizing and

iterating. The linearization is accomplished by using the previous iterate for the coefficient $V = K - \phi_x^0$. The resulting linear system is tridiagonal and is solved by optimum Gaussian elimination. The column iteration process is terminated when the difference between successive iterates is less than an arbitrary small amount (usually 10^{-5}). As in the 2-D case, convergence is usually achieved in three or four iterations. In the unsteady solution it is recalled that the equation is linear so that no column iteration is required.

After each column is solved, it is relaxed using a variable relaxation factor which depends on the local nature of the equation; $\omega \sim 1.7$ for elliptic points and $\omega \sim .75$ for hyperbolic points. The column solution process is performed for each column in turn sweeping the grid from left to right in x and from the wing root (k = 1) to the farfield (k = KM) in z. The entire grid is swept repeatedly in this manner until the change in ϕ for all grid points during one grid sweep is less than some arbitrary small amount.

The numerical treatment of airfoil and wake boundary conditions in both steady and unsteady cases is the same as the 2-D case with the exception that the airfoil shape function is now a function of z as well as x and the airfoil circulation is a function of z along the airfoil. In the subsonic freestream case considered to date, asymptotic solutions for the steady and unsteady systems described in Section 3.2 are used to fix a Dirichlet boundary condition on five sides of the grid. On the grid boundary containing the wing root, a symmetry boundary condition is used whereby $\phi_z \equiv 0$ on z = 0. The farfield solution depends on the spanwise distribution of circulation and as the solution for circulation is refined the farfield is updated periodically during the solution process.

The solution process summarized above has worked well in the few cases calculated to date. Convergence, for instance, seems to be comparable to the two-dimensional method as will be discussed in Section 6.0. It is reiterated that the details of the finite difference equations and wing, wake and farfield boundary conditions as well as details of the iteration procedures are identical in TDSTRN and TDUTRN as described previously for STRANS and UTRANS in Reference 2.

3.2 Steady and Unsteady Farfield Prescriptions

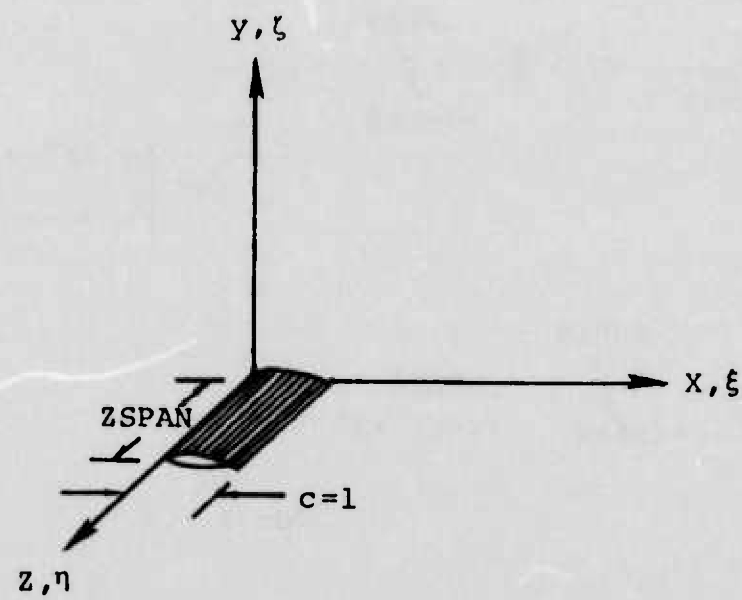

FIGURE 4. COORDINATE DEFINITION

The development of three-dimensional subsonic farfield approximations for the steady and unsteady perturbation potentials proceeds in the same manner as described in previous work for the two-dimensional flow. As before the method involves the approximation of various integrals over the wing and wake which result from the application of Green's theorem to the appropriate partial differential equation. Klunker[7] has used the method to develop asymptotic solutions for the three-dimensional steady flow and his result in the following form is used:

$$\phi^0_{ff}(x,y,z) = -\frac{x}{2\pi R^3}\left[\int_{-ZSPAN}^{ZSPAN}\int_0^1 t(\xi,\eta)\,d\xi\,d\eta\right]$$

$$+\begin{cases} \dfrac{y}{4\pi(y^2+z^2)}\left(1+\dfrac{x}{R}\right)\displaystyle\int_{-ZSPAN}^{ZSPAN}\gamma(\eta)\,d\eta & \text{for } \begin{cases} y^2+z^2 \to \infty \\ x \to -\infty \end{cases} \\ \dfrac{y}{2\pi}\displaystyle\int_{-ZSPAN}^{ZSPAN}\dfrac{\gamma(\eta)}{(z-\eta)^2+y^2}\,d\eta & \text{for } x \to +\infty \end{cases} \qquad (14)$$

where $R = [(x^2+K(y^2+z^2)]^{\frac{1}{2}}$, $t(\xi,\eta)$ is the wing thickness distribution and $\gamma(\eta)$ is the spanwise distribution of circulation.

The development of an asymptotic solution for the unsteady perturbation potential follows the method of Klunker and is now described in some detail. The field equation for the unsteady perturbation potential (Equation 10) is written as:

$$L[\phi] \equiv K\phi_{xx} + \phi_{yy} + \phi_{zz} - 2i\Omega\phi_x + k\Omega\phi$$
$$= (\phi^0_x\,\phi_x)_x \qquad (15)$$

The application of Green's theorem to the linear operator L and the use of wing and wake boundary conditions and weak shock conditions results in the following integral equation for ϕ:

$$\phi(x,y,z) = \iint\limits_{wing} \Delta\phi(\xi,\eta)\ \psi_\zeta\ d\xi d\eta$$

$$\underbrace{+ \int\limits_{SPAN} \gamma(\eta) \int_1^\infty \psi_\zeta\ e^{-ik(\xi-1)} d\xi d\eta}_{\text{wake integral}} \qquad (16)$$

$$+ \iiint_{-\infty}^{\infty} (\phi_\xi^0 \phi_\xi^0)\ \psi_\zeta\ d\xi d\eta d\zeta$$

where ψ is the source solution to $L[\Phi] = 0$:

$$\psi(x,y,z;\xi,\zeta,\eta) = \frac{1}{4\pi R}\, e^{\, i\left(\frac{\Omega}{K}(x-\xi) - \frac{\mu}{\sqrt{K}} R\right)} \qquad (17)$$

where

$$\mu = \sqrt{\Omega\left(\frac{\Omega}{K} + k\right)}$$

$$R = \sqrt{(x-\xi)^2 + K\left[(y-\zeta)^2 + (z-\eta)^2\right]}$$

The use of the source function ψ in Equation 16, neglecting the volume integral as a higher order term, and after considerable manipulation and approximation (as $x^2+y^2+z^2 \to \infty$) of the various integrals results in the following farfield solution:

$$\phi_{ff}(x,y,z) = \frac{Ky}{4} \frac{(1+i\frac{\mu}{\sqrt{K}}R)}{R^3} e^{i\left(\frac{\Omega}{k}x - \frac{\mu}{\sqrt{K}}R\right)} \int_{-ZSPAN}^{ZSPAN}\int_0^1 \Delta\phi(\xi,\eta) e^{-i\frac{\Omega}{K}\xi} d\xi d\eta$$

$$+ \frac{y}{4\pi} e^{-ik(x-1)} \cdot \begin{cases} \left[G_1(x,y,z;\eta)+G_2(x,y,z;\eta)\right]_{\eta=0} \displaystyle\int_{-ZSPAN}^{ZSPAN} \gamma(\eta)d\eta & \text{for } \begin{cases} y^2+z^2\to\infty \\ x\to-\infty \end{cases} \\ \displaystyle\int_{-ZSPAN}^{ZSPAN} \left[G_1(x,y,z;\eta)+G_2(x,y,z;\eta)\right] \gamma(\eta)d\eta & \text{for } x\to+\infty \end{cases} \qquad 13)$$

where

$$G_1 = \frac{KM_\infty e^{-ikt_1}}{R[R-M_\infty(x-1)]}$$

$$G_2 = \frac{I_1}{r^2}$$

and where

$$r = \sqrt{K[y^2+(z-\eta)^2]}$$

$$R = \sqrt{(x-1)^2+K[y^2+(z-\eta)^2]}$$

$$t_1 = \frac{M_\infty R-(x-1)}{1-M_\infty^2}$$

I_1 in the equation, is an integral that can be evaluated using a rational approximation to its integrand and is presented at the end of the section.

It is noted that both the steady and unsteady farfield solutions involve integrals over the wing and span which depend on the solution $(\Delta\phi, \gamma)$. These integrals are evaluated numerically as the numerical solution proceeds and the respective equations are used to update the values of the steady or unsteady potential on the farfield boundaries.

The use of a rational approximation to evaluate the portion of the wake integral given as I_1 above, results in the following function:

$$I_1 = \frac{-|u_1|}{\sqrt{1+u_1^2}} e^{-ik\hat{r}|u_1|} + ik\hat{r} \sum_{\nu=0}^{11} \frac{b_\nu}{c\nu+ik\hat{r}} e^{-(c\nu+ik\hat{r})|u_1|}$$

$$+ \left(1 - \frac{u_1}{|u_1|}\right) \cdot \mathrm{Re} \left\{ \frac{u}{\sqrt{1+u^2}} e^{-ik\hat{r}u} \Bigg|_0^{|u_1|} \right. \tag{19}$$

$$\left. - ik\hat{r} \sum_{\nu=0}^{11} \frac{b_\nu}{c\nu+ik\hat{r}} e^{-(c\nu+ik\hat{r})u} \Bigg|_0^{|u_1|} \right\}$$

where

$$\hat{r} = \frac{r}{[(1+\gamma)\delta M_\infty^2]^{1/3}}$$

$$u_1 = \frac{t_1}{\hat{r}}$$

and c = 0.372 with b_ν defined in the table.

ν	b_ν
0	1.0
1	-0.2418 6198
2	2.7918 027
3	-24.9910 79
4	111.5919 6
5	-271.4354 9
6	305.7528 8
7	41.1836 30
8	-545.9853 7
9	-644.7815 5
10	-328.7275 5
11	64.2795 11

4.0 PROGRAM DESCRIPTIONS

Computer programs TDSTRN and TDUTRN, used in conjunction, implement the theory and numerical solution procedure for unsteady transonic flow described in the previous two sections. As described above, the boundary value problems for the steady perturbation potential (Equation 9) and the unsteady perturbation potential (Equation 10) are solved in TDSTRN and TDUTRN respectively using a finite difference relaxation procedure. The use and manipulation of magnetic tapes forms an integral part of the operation of each program as well as serving as the necessary "data link" between the two programs. As a result the user is assumed to have some familiarity with the use of tapes and their manipulation with control cards. The reading and writing of data files on magnetic tape is described in the next section and motivated in Section 6.0. In this section, the logical flow of the TDSTRN and TDUTRN programs is described and a brief summary of each subroutine is presented. Both programs are quite similar in logical approach and operation, so that they are described together. Differences between the programs are highlighted with appropriate comments as needed.

The logical flow of the TDSTRN and TDUTRN programs are almost identical with minor exceptions noted in the description below. The calculation is begun by reading card input and, if a restart is being performed, a tape dump. In TDUTRN the tape dump of the steady solution being perturbed is also read. All finite difference coefficients and airfoil boundary conditions are initialized in a call to INITAL and subsonic farfield quantities are initialized in a call to FARFLD. If a restart is not being performed, initial values for ϕ at all grid points are determined by the linearized subsonic or supersonic solution. The computational cycle is executed by setting up the tridiagonal equations for a column of grid points using the mixed differencing finite difference equations. The equations are solved iteratively in TDSTRN and in one pass in TDUTRN, by Gaussian elimination in a call to TRI. Each column is solved and relaxed in turn proceding through the grid from left to right. The grid is swept iteratively in this manner until the change in ϕ for all grid points is less than EPSGRD(1). A call to PRINT prints out the airfoil pressure coefficients every NPRINT iterations, and the farfield is updated every NGFF iterations, in FARFLD.

When the converged solution is obtained, a tape dump of all relevant input and calculated quantities is performed and a call to FPRINT calculates and prints out the airfoil pressure and force coefficients. Various diagnostic prints are also performed in TDSTRN and TDUTRN after every grid iteration, when the farfield is updated, when the grid is refined and when a tape dump is performed. The iterative procedure may also be terminated when the maximum number of iterations (NGRID) has been exceeded. In either case, a final tape dump and final print are executed.

A summary of each subroutine is now presented.

TDSTRN/TDUTRN

These are the driver routines for the respective programs. The logical flow of the mixed differencing relaxation procedure as just described is controlled by these routines and all operations including input, initialization, finite difference solution and output are performed either internally or by calls to the various subroutines described below.

DØUBLE

(Not in present version).

FARFLD

The subsonic farfield is calculated and updated in this routine using the asymptotic solutions for the steady or unsteady perturbation potentials.

FLP (in TDSTRN only)

This is a function statement which contains the airfoil lower surface slope distribution used in the linearized tangency boundary condition. This function is called from subroutine INITAL and its value at each grid point on the lower surface of the airfoil is stored in the FPL array.

FPRINT

This routine produces the final print and is called when the solution has converged to the desired accuracy or when the problem is terminated for reaching the maximum number of grid iterations allowed (NGRID). The unscaled pressure coefficients above and below the airfoil at various specified spanwise stations and the airfoil force coefficients are also calculated and printed out in this routine.

FPU (in TDSTRN only)

This is a function statement which contains the airfoil upper surface slope distribution used in the linearized tangency boundary condition. This function is called from subroutine INITAL and its value at each grid point on the upper surface of the airfoil is stored in the FPU array. The doublet strength due to airfoil thickness (DØUB) must also be given in this subroutine. This quantity is defined by an integral of the airfoil thickness distribution function (normalized to airfoil thickness):

$$\mathrm{D\emptyset UB} = \int_{-\mathrm{ZSPAN}}^{+\mathrm{ZSPAN}} \int_{0}^{1} t(\xi,\eta)\,d\xi\,d\eta$$

GAMFUN

This routine performs the relaxation to update farfield circulation (GAMFF).

INITAL

The finite difference coefficients AX1, AX2, BX1, BX2, CX, AY1, AY2, AZ1, AZ2, ΔX(DX), ΔY(DY) and ΔZ(DZ) are computed in this subroutine. The airfoil boundary conditions FPU and FPL are also set here, using functions FUP and FLP respectively.

PRINT

This routine computes and prints the scaled pressure coefficients above and below the airfoil every NPRINT grid iterations.

TRI

This routine solves a system of tridiagonal equations using Gaussian elimination.

WAKE

This routine solves an integral used in the unsteady farfield solution based on a rational approximation for the integrand.

5.0 INPUT AND OUTPUT

A description of the input required to run TDSTRN and TDUTRN, and the resulting output of each program is presented in this section. All card input is entered using the standard CDC NAMELIST package with the exception of a title card.

5.1 TDSTRN Input

The input for TDSTRN is now considered in three sets. Recommended and/or typical values for some of the input variables which control the numerical scheme, appear in parentheses. Also presented at the end of this section is a description of the restart capability which requires input from a magnetic tape dump of a previous calculation.

First Set

BCD title card containing any information in columns 1 through 80 (Format 8A10). This can be used to define the case being run and is printed out on the last page of output which presents the final converged results.

Second Set

The second set of data is read in under NAMELIST name $CØNTRL. The single variable read defines the use of the restart option. Some comments concerning the mechanics of the use of this option are given at the end of the section.

NAME	DESCRIPTION
ITAPE	This is a flag for using a restart tape. ITAPE = 0 means the problem is being started from scratch (iteration 0) using an initial guess defined in TDSTRN. ITAPE = 1 means the problem is being restarted from a previous run which is to be read from a dump tape.

Third Set

The third set of data is read in under NAMELIST name $IN, and includes all of the variables required to define a problem, and control the numerical iteration procedure.

NAME	DESCRIPTION
X	An Array containing the streamwise grid coordinates; IM of them
Y	An array containing the normal grid coordinates; JM of them
Z	An array containing the spanwise grid coordinates; KM of them
IM	Number of grid points in the streamwise direction (maximum of 40)
JM	Number of grid points in the normal direction (maximum of 40)
KM	Number of grid points in the spanwise direction; (maximum of 20)
ILE	I location of airfoil leading edge (X(ILE))
ITE	I location of airfoil trailing edge (X(ITE))
JW	J location of airfoil (Y(JW))
KSPAN	K location of wing tip
ZSPAN	Wing semi-span; Z location of wing tip
M8	Freestream Mach number
GAM	γ, ratio of specific heats
DEL	Airfoil thickness ratio in percent
ALPHA	Airfoil angle of attack in radians
GAMFF	Initial guess for the spanwise distribution of airfoil circulation; to be used in the initialization of the farfield; KSPAN values.
NGFF	Every NGFF grid iterations the farfield is updated ($\sim$10).

NAME	DESCRIPTION
ØMEGAH	Relaxation parameter for hyperbolic grid points ($\sim$.75)
ØMEGAE	Relaxation parameter for elliptic grid points ($\sim$1.7)
ØMEGAP	Relaxation parameter for parabolic grid points ($\sim$.75)
EPSCØL	Convergence criteria for column solution. The change in $\dot{\phi}^0$ during a column iteration at every point in the column must be less than EPSCØL for convergence to occur ($\sim 5 \times 10^{-5}$)
NCØL	Maximum number of column iterations allowed. Note that if NCØL iterations is reached without convergence, a printout of the degree of convergence is given and the calculations proceed as if convergence had occurred ($\sim$10)
EPSGRD	An array containing criteria to control grid convergence. The change in ϕ^0 at every grid point during one grid sweep must be less than EPSGRD(1) for convergence to occur.
KEPS	Set equal to 1. (Not used in current version)
NGRID	Maximum number of grid iterations allowed. When the number of grid iterations equals NGRID the calculation is terminated and a final print given.
NDUMP	Binary tape dump frequency. Every NDUMP grid iterations current values of all variables will be dumped on tape. Note that a tape dump occurs automatically whenever the grid converges or the number of grid iterations equals NGRID (set equal to large number if a dump of only the final iteration is desired).
NPRINT	Every NPRINT grid iterations the scaled pressure coefficient above and below the airfoil is printed.

NAME	DESCRIPTION
IK	Setting IK = 1, flags the use of a previous solution as an initial guess for the current problem where the Mach number, airfoil thickness or shape and/or angle of attack may be different.
NKPRT	Number of spanwise sections for which pressure coefficient data is printed in final print.
KPRT	K location of spanwise sections for which pressure coefficient data is printed in final print (maximum of 20).
ZE	Spanwise locations for numerical integration along span used in farfield calculations (maximum of 25). This permits the specification of more spanwise points than available in grid (KSPAN) to increase accuracy of the numerical evluation of wing integrals.
NZE	Number of spanwise locations for wing integration.

The input data listed above are necessary to initiate a calculation for which no previous calculation is available. Most calculations, however, are performed as restarts using data which has been stored as binary files on the restart tape according to the format described in Section 5.2. This use of the restart capability is an inherent aspect of the recommended computational procedure. Some brief comments describing the initiation of a calculation using the restart capability are pertinent at this juncture.

It is noted that the restart or dump tape (TAPE7) may be manipulated in any way desired using the appropriate control cards. In general the tape will contain data from many runs, stored as individual binary files. For restarting the TDSTRN program, the desired file from the restart tape (TAPE7) is copied to a disc file (TAPE8). The user is reminded to rewind TAPE8. TAPE7 is then positioned at the end of the last file on the tape so that new dumps can be written by the program without losing any of the old data. The first two sets of data are than input with ITAPE=1. In the third set of data the following control variables are needed as input:

ØMEGAH, ØMEGAE, ØMEGAP, EPSCØL, EPSGRD, NDUMP, NCØL, NGRID, NGFF, PGFF, KEPS, NPRINT, NKPRT, KPRT, ZE, and NZE.

The remaining input variables are stored on the restart tape and need not be input unless the restart option is being used to run a new case. If a new case is being run, IK must be set to 1 which allows the Mach number, airfoil thickness or airfoil angle of attack (M8, DEL, ALPHA) to be changed. If the airfoil angle of attack and/or the flap angle are changed a new guess for the farfield circulation (GAMFF) can and should be made.

5.2 TDSTRN Output

The output from TDSTRN consists of three parts: (i) a continuous commentary which describes the progress of the iterative solution procedure, (ii) a final print summarizing results of interest from the final converged solution, and (iii) a binary tape dump of all pertinent input and calculated parameters.

The continuous commentary consists of various print statements executed in the main program TDSTRN or the subroutine PRINT which describe the current state of the solution as well as the occurrence of various "milestones" in the iteration process. The only print that occurs every iteration is the value of the maximum change in ϕ^0 throughout the grid during one grid iteration. When a column iteration fails to converge, a print occurs which defines the degree of column convergence and the j and k locations of the most poorly converged point. Every NGFF iterations, the subsonic farfield is updated and the new values of farfield circulation (GAMFF) and airfoil circulation (GAMTE) are printed. The user can examine the effect of degree of convergence on the solution by specifying a print of the scaled pressure coefficients on the upper and lower airfoil surfaces every NPRINT iterations. Finally, a descriptive print occurs at certain milestone points such as the occurrence of a binary tape dump and solution convergence.

The final print is executed in subroutine FPRINT when the solution has converged to the desired accuracy or when the number of grid iterations equals NGRID. The print is self-explanatory and includes the input parameters which define the problem and various calculated quantities of interest. The calculated quantities are of course based on the final converged solution. The section lift coefficients are printed out as well as the upper and lower surface pressure coefficients for various spanwise coordinates.

The most important form of TDSTRN output is the binary tape dump of all input parameters defining the problem and of the most recent values of ϕ^0 at all grid points. A tape dump occurs automatically if the solution has converged to the desired accuracy or if the number of grid iterations equals NGRID. The user may also specify that such a dump occur every NDUMP grid iterations. The tape so generated, not only forms a permanent record of the results of a calculation for possible future editing and examination but also forms a necessary part of the computational procedure. Most important is its use as required input for a TDUTRN calculation. However, it may also be used to restart the calculation to refine accuracy or convergence or be used as the initial guess for ϕ^0 throughout the grid for a similar calculation, as described in Section 5.1.

The format used for writing and reading the binary tape is given in the following FORTRAN statements:

```
WRITE (7)    NITERG,IM,IM1,JM,JM1,KM,KM1,JW,
             JWP1,JWM1,ITE,ILE,KSPAN,KCAP,DEL,
             ALPHA,NDB,M8,GAM,DYBU1,DYBU2,DYBL1,
             DYBL2,DØUB,ZSPAN

WRITE (7)    (X(I),DX(I),AX1(I),AX2(I),BX1(I),
             BX2(I),CX(I),I=1,IM)

WRITE (7)    (Y(I),DY(I),AY1(I),AY2(I),I=1,JM)

WRITE (7)    (Z(I),DZ(I),AZ1(I),AZ2(I),I=1,KM)

L=ITE*KM

WRITE (7)    (FPU(I),FPL(I),PHIUB(I),I=1,L)

WRITE (7)    (GAMTE(I),GAMFF(I),I=1,KSPAN)

L=IM*JM*KM

WRITE (7)    (PHI(I),I=1,L)

END FILE 7
```

Any information may be retrieved from the tape by using the appropriate READ statements as is done in the restart option described above.

5.3 TDUTRN Input

The input for TDUTRN consists of normal card input plus input from a binary file which contains the steady solution generated by an TDSTRN run. TDUTRN also has a restart capability which is implemented in exactly the same manner as previously described in Section 5.1 for TDUTRN and elaborated upon at the end of this section. The required input is now described and some comments are presented at the end of this section pertaining to the tape read of the steady solution. As before, the card input is described in three sets.

First Set

BCD title card containing any information in columns 1 through 80 (Format 8A10).

Second Set

The second set of data is read in under NAMELIST name $CØNTRL.

NAME	DESCRIPTION
ITAPE	This is a flag for using a restart tape. ITAPE=0 means the problem is being started from scratch (iteration 0), ITAPE=1 means the problem is being restarted from a previous run using the restart tape. Note that a tape is also used for the input of steady results independent of the value of ITAPE.

Third Set

The third set of data is read in under NAMELIST name $IN.

NAME	DESCRIPTION
X	An array containing the streamwise grid coordinates; IM of them.
Y	An array containing the normal grid coordinates; JM of them.
Z	An array containing the spanwise grid coordinate KM of them

NAME	DESCRIPTION
IM	Number of grid points in the streamwise direction (maximum of 40).
JM	Number of grid points in the normal direction (maximum of 40).
KM	Number of grid points in the spanwise direction (maximum of 20).
ILE	I location of airfoil leading edge (X(ILE)).
ITE	I location of airfoil trailing edge (X(ITE)).
JW	J location of airfoil (Y(JW)).
SMALLK	Reduced frequency based on chord = $\omega c/U$.
KSPAN	K location of wing tip
GAMFF	Initial guess for the airfoil circulation used in the intialization of the farfield. Note that GAMFF is a complex number.
NGFF	Every NGFF grid iterations the airfoil circulation in the farfield is updated. This also causes the farfield to be updated ($\sim$10).
PGFF	Relaxation parameter used in the iteration for the airfoil circulation in the farfield ($\sim$1.5).
ØMEGAH	Relaxation parameter for hyperbolic grid points ($\sim$.75).
ØMEGAE	Relaxation parameter for elliptic grid points ($\sim$1.7).
ØMEGAP	Relaxation parameter for parabolic grid points ($\sim$.75).
EPSGRD	An array containing criteria to control grid convergence. The change in ϕ^1 at every grid point during one grid sweep must be less than EPSGRD(1) for convergence to occur.
KEPS	Set equal to 1. (Not used in current version)

NAME	DESCRIPTION
NGRID	Maximum number of grid iterations allowed. When the number of grid iterations equals NGRID the calculation is terminated.
NDUMP	Binary tape dump frequency. Every NDUMP grid iterations current values of all variables will be dumped on tape. Note that a tape dump occurs automatically whenever the grid converges or the number of grid iterations equals NGRID. (Set equal to large number if a dump of only the final iteration is desired.)
NPRINT	Every NPRINT grid iterations the scaled upper and lower surface pressure coefficient per unit angle of oscillation is printed.
IK	Setting IK=1 allows the user to use a previous solution as an initial guess for the current problem where the reduced frequency and/or mode of oscillation is different.
XP	Steamwise location of pitch point for pitching oscillation.
ITYPE	Unsteady mode of rigid body oscillation ITYPE=1 → Pitch about XP ITYPE=3 → Uniform plunge
IØPT	Unsteady formulation option; IØPT=0 for low frequency approximation, IØPT=1 for general frequency theory.
NKPRT	Number of spanwise sections for which pressure coefficient data is printed in final print.
KPRT	K location of spanwise sections for which pressure coefficient data is printed in final print (maximum of 20).
ZE	Spanwise locations for numerical integration along span used in farfield calculation (maximum of 20). This permits the specification of more spanwise points than available in grid (KSPAN) to increase accuracy of the numerical integration of wing integrals.
NZE	Number of spanwise locations for wing integration.

It is recalled, that the solution for the unsteady perturbation potential, implemented in TDUTRN, requires the solution of the steady potential, generated by TDSTRN. This is accomplished by reading the appropriate file on a dump tape generated by TDSTRN, in much the same way as is done in the restart option. It is instructive to briefly describe the TDUTRN restart including the tape read of the steady solution.

Restarting the TDUTRN program is only slightly more complicated than TDSTRN. In this case, two tape dumps or files are required. First the file containing the desired steady tape dump is copied from TAPE7 to a disc file, TAPE8. Next the file containing the desired unsteady tape dump is copied from TAPE7 to a disc file, TAPE9, and TAPE8 and TAPE9 are rewound for reading by the program. TAPE7 is then positioned at the end of the last file on the tape in preparation for accepting a new tape dump. The first two sets of data are input as before (be sure to set ITAPE=1). In the third set of data the following variables are necessary:

> ØMEGAH, ØMEGAE, ØMEGAP, EPSGRD, NDUMP, NGRID, NGFF, PFGG, KEPS, NPRINT, NKPRT, KPRT, ITYPE, IØPT, ZE, and NZE.

Again there is an option (IK=1) which allows the user to change the reduced frequency and/or the mode of oscillation (SMALLK and ITYPE). In either case a new guess for the farfield circulation (GAMFF) should be made.

5.4 TDUTRN Output

The output from TDUTRN is very similar to that of TDSTRN and includes a continuous commentary, final print and binary tape dump.

The printed output is of the form described above for TDSTRN. The only difference is that the field variables in TDUTRN are complex so that the real and imaginary parts are printed out in that order. The descriptive prints are all the same with the deletion of the unneeded comment on column convergence. The final print is executed in subroutine FPRINT when the solution has converged or has reached the maximum number of iterations desired by the user (NGRID). The print includes all important input variables which define both the steady solution being perturbed and the unsteady solution being generated. Also, various calculated quantities, based on the final con-

verged solution, are printed. These include the real and imaginary parts of the unsteady contribution (per unit angle of oscillation) to the aerodynamic force coefficients. Also unsteady contributions to the upper and lower surface pressure coefficients (per unit angle of oscillation) are printed for every computational point on the airfoil. It is again noted that these are complex so that the real and imaginary parts are printed out in order.

The other form of TDUTRN output is the binary tape dump of all input parameters and the most recent values of ($Re\phi^1$, $Im\phi^1$) at all grid points. As before the tape dump occurs automatically at normal program termination or at the users discretion every NDUMP iterations. The format used for writing and reading the binary tape is given in the following FORTRAN statements:

```
WRITE (7)     NITERG,IM,IM1,JM,JM1,KM,KM1,JWPI,
              JWM1,ILE,ITE,KSPAN,ØMEG,SMALLK,
              DYBU1,DYBU2,DYBL1,DYBL2,NDØUB,XP

WRITE (7)     (X(I),DX(I),AX1(I),AX2(I),BX1(I),
              BX2(I),CX(I),I=1,IM)

WRITE (7)     (Y(I),DY(I),AY1(I),AY2(I),I=1,JM)

WRITE (7)     (Z(I),DZ(I),AZ1(I),AZ2(I),I=1,KM)

L=ITE*KM

WRITE (7)     (FPU(I),FPL(I),PHIUB(I),I=1,L)

WRITE (7)     (GAMTE(I),GAMFF(I),I=1,KSPAN)

L=IM*JM*KM

WRITE (7)     (PHI(I),I=1,L)

END FILE 7
```

6.0 PROGRAM USAGE

The general structure and usage of the three-dimensional programs TDSTRN and TDUTRN are very similar to that for the original two-dimensional versions STRANS and UTRANS. This being the case, it is recommended for economy sake that the first time user initially become acquainted with those programs. The programs are documented in detail in Reference 2 so that the comments concerning program usage in this manual are kept necessarily brief.

In their present configuration, both TDSTRN and TDUTRN allow a maximum of 11,500 computational grid points and the number of grid lines in the streamwise, normal and spanwise directions must each be less than 40, 40, 20 respectively. In this configuration, TDSTRN requires 70.5_8K words to load and 57.0_8K words to execute and TDUTRN requires 161.7_8K words to load and 150.0_8K words to execute. This configuration was chosen so that each program could fit into small core storage of a CDC 7600 computer. If greater storage is available and used, (Ex. CDC 6600) it is a relatively simple matter to increase the array sizes of the primary variables PHI, X,Y, Z, FPU, FPL, etc.

Detailed comments and suggestions are given in Reference 2 concerning grid design, farfield location and update, choice of relaxation factors and accuracy and convergence. These same comments apply to the present three-dimensional programs and will not be repeated here. The sample cases presented in the next section should provide some guidance with respect to such items.

7.0 SAMPLE CASES

Detailed input and sample output for sequences of TDSTRN and TDUTRN runs are presented in this section.

7.1 TDSTRN Test Case

A sequence of computer runs are described in this section, which calculate the steady transonic flow over a 6 percent thick, symmetric circular arc, rectangular plan-form wing with aspect ratio 8, at $M_\infty = .86$, $\alpha = 0$. The individual runs required to complete the calculation are described in the run log given in Table 1. The table lists the restart tape read by each run, total grid iterations, convergence achieved and the tape dump generated. The grid used consisted of approximately 11000 points with IM=30 over $-3.2 \leq x \leq 3.4$, JM=19 over $-5.4 \leq y \leq 5.4$ and KM=19 over $0 \leq Z \leq 6.0$. In the x direction, 16 grid lines were distributed along the airfoil chord with $\Delta X \sim .06$ and in the Z direction 10 grid lines were distributed over the span with $\Delta Z \sim .2$. The runs shown in the log implement a "boot-strapping" technique by which the calculation is initiated at a low sub-critical Mach number and the Mach number raised in later runs to the final desired value. The final run was taken to a convergence of $\Delta\phi_{max} = 3.7 \times 10^{-5}$. All runs were completed in a total time of 65 seconds on a CDC 7600 which indicates a computer time requirement of $3. \times 10^{-5}$ CPU sec/grid point/iteration. The final convergence achieved is believed to be more than sufficient for engineering accuracy.

Run	M_∞	Restart Tape Used	Grid Iterations	Convergence Achieved	Tape Dump Generated
1S	.7	--	38	10^{-3}	1S
2S	.8	1S	19	10^{-3}	2S
3S	.86	2S	50	1.9×10^{-4}	3S
4S	.86	3S	50	3.7×10^{-5}	4S

TABLE 1. SEQUENCE OF RUNS FOR TDSTRN SAMPLE CASE

7.1.1 Input for TDSTRN Sample Cases

The card input for each of the TDSTRN runs described above is given in this section.

- Run 1S: no tape read, generate file 1S

```
                    ***3D CIRCULAR ARC***

$CØNTRL
ITAPE=0,
$END
$IN
X(1)=-3.2,-2.2,-1.5,-1.02,-.67,-.42,-.24,-.1,0.,.07,
    .14,.21,.28,.35,.42,.5,.55,.6,.65,.7,.76,.82,
    .9,1.,1.14,1.34,1.62,2.02,2.58,3.38,
Y(1)=-5.4,-3.41,-2.91,-1.91,-1.21,-.74,-.43,-.22,-.08,
    0.,.08,.22,.43,.74,1.21,1.91,2.91,3.41,4.3,
Z(1)=0.,.25,.5,.75,1.,1.25,1.5,1.75,1.9,2.,2.1,2.25,
    2.45,2.75,3.2,3.85,4.75,6.,6.8,
IM=30,
JM=19,
KM=19,
ILE=9,
ITE=24,
JW=10,
KSPAN=10,
ZSPAN=2.,
M8=.7,
GAM=1.4,
DEL=.06,
ALPHA=0.0,
GAMFF(1)=10*0.,
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSCØL=5.E-5,
EPSGRD(1)=1.E-3,
NDUMP=2000,
NCØL=10,
NGRID=50,
NGFF=2000,
PGFF=1.5,
KEPS=1,
IK=0,
NPRINT=5,
NKPRT=10,
KPRT(1)=1,2,3,4,5,6,7,8,9,10,
ZE(1)=0.,2.,
NZE=2,
$END
```

- Run 2S: read file 1S, generate file 2S

```
***3D CIRCULAR ARC***

$CØNTRL
ITAPE=1,
$END
$IN
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSCØL=5.E-5,
EPSGRD=1.E-3,
NDUMP=2000,
NCØL=10,
NGRID=50,
NGFF=2000,
PGFF=1.5,
KEPS=1,
NPRINT=5,
NKPRT=10,
KPRT(1)=1,2,3,4,5,6,7,8,9,10,
ZE(1)=0.0,2.0,
NZE=2,
IK=1,
M8=0.8,
$END
```

- Run 3S: read file 2S, generate file 3S

```
***3D CIRCULAR ARC***

$CØNTRL
ITAPE=1,
$END
$IN
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSCØL=5.E-5,
EPSGRD=1.E-4,
NDUMP=2000,
NCØL=10,
NGRID=50,
NGFF=2000,
PGFF=1.5
KEPS=1,
NPRINT=5,
NKPRT=10,
```

```
KPRT(1)=1,2,3,4,5,6,7,8,9,10,
ZE(1)=0.0,2.0,
NZE=2,
IK=1,
M8=.86,
$END
```

- Run 4S: read file 3S; generate file 4S

```
                  *** 3D CIRCULAR ARC***

$CØNTRL
ITAPE=1,
$END
$IN
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSCØL=5.E-5,
EPSGRD=1.E-3,
NDUMP=2000,
NCØL=10,
NGRID=50,
NGFF=2000,
PGFF=1.5,
KEPS=1,
NPRINT=5,
NKPRT=10,
KPRT(1)=1,2,3,4,5,6,7,8,9,10,
ZE=0.0,2.0,
NZE(1)=2,
IK=0,
$END
```

7.1.2 Sample Output for TDSTRN Test Case

The following pages contain a sample of the continuous commentary output for the first 4 cycles of Run 1S in addition to the final printed page of all runs. Also included is the complete final output for the final converged result (Run 4S).

● SAMPLE OUTPUT FROM RUN 1S

```
SIMILARITY PARAMETER (K) =   .29867E+01
SCALING FACTOR (CP/CPBAR) =   .18520E+00

AT ITERATION    1 COLUMN  11 FAILED TO CONVERGE ERR =  -.37921E-03 J =  7 K =  1

AT ITERATION    1 COLUMN  15 FAILED TO CONVERGE ERR =  -.12937E-03 J =  6 K =  1

AT ITERATION    1 COLUMN  22 FAILED TO CONVERGE ERR =  -.15179E-03 J =  9 K =  1

AT ITERATION    1 COLUMN  25 FAILED TO CONVERGE ERR =  -.73166E-04 J =  8 K =  1

AT ITERATION    1 COLUMN   9 FAILED TO CONVERGE ERR =   .72460E-04 J =  9 K =  2

AT ITERATION    1 THE MAXIMUM ERROR =  -.23879E+01 AND OCCURRED AT NODE  334

AT ITERATION    2 COLUMN   9 FAILED TO CONVERGE ERR =  -.66027E-04 J =  8 K =  1

AT ITERATION    2 COLUMN  12 FAILED TO CONVERGE ERR =   .83003E-04 J =  7 K =  1

AT ITERATION    2 COLUMN  15 FAILED TO CONVERGE ERR =  -.59371E-04 J =  7 K =  1

AT ITERATION    2 THE MAXIMUM ERROR =   .22038E+01 AND OCCURRED AT NODE  278

AT ITERATION    3 COLUMN  19 FAILED TO CONVERGE ERR =   .25477E-02 J =  9 K =  1

AT ITERATION    3 COLUMN  20 FAILED TO CONVERGE ERR =  -.27265E-03 J =  9 K =  1

AT ITERATION    3 COLUMN  22 FAILED TO CONVERGE ERR =  -.74079E-04 J =  8 K =  1

AT ITERATION    3 COLUMN  25 FAILED TO CONVERGE ERR =  -.14493E-03 J =  9 K =  1

AT ITERATION    3 THE MAXIMUM ERROR =   .30739E+01 AND OCCURRED AT NODE  410

AT ITERATION    4 COLUMN  14 FAILED TO CONVERGE ERR =   .68272E-04 J =  6 K =  1

AT ITERATION    4 COLUMN  25 FAILED TO CONVERGE ERR =  -.72433E-04 J =  8 K =  1

AT ITERATION    4 COLUMN  24 FAILED TO CONVERGE ERR =  -.24118E-03 J =  8 K =  2

AT ITERATION    4 COLUMN  25 FAILED TO CONVERGE ERR =  -.31249E-03 J =  6 K =  2

AT ITERATION    4 THE MAXIMUM ERROR =   .44832E+01 AND OCCURRED AT NODE 1036

AT ITERATION    4 AND K =  1 SCALED PRESSURE COEFFICIENT, UPPER (ILE TO ITE) =
  .16892E+01   .16314E+01   .94405E+00   .32156E+01   .18217E+01   .15594E+02   .30448E+02   .75677E+01  -.16345E+01  -.69093E+01
 -.26390E+02  -.44045E+02  -.55156E+02  -.54553E+02  -.41899E+02  -.12317E+02

AT ITERATION    4 AND K =  1 SCALED PRESSURE COEFFICIENT, LOWER (ILE TO ITE) =
  .16892E+01   .16314E+01   .94405E+00   .32156E+01   .18217E+01   .15594E+02   .30448E+02   .75677E+01  -.16345E+01  -.69093E+01
 -.26390E+02  -.44045E+02  -.55156E+02  -.54553E+02  -.41899E+02  -.12317E+02

AT ITERATION    4 AND K = 10 SCALED PRESSURE COEFFICIENT, UPPER (ILE TO ITE) =
  .64128E+00   .20246E+00  -.62959E-01  -.33531E+00  -.51373E+00  -.56171E+00  -.62432E+00  -.64968E+00  -.68677E+00  -.79599E+00
 -.10038E+01  -.13154E+01  -.11538E+01  -.68828E+00  -.36427E+00   .28329E+00

AT ITERATION    4 AND K = 10 SCALED PRESSURE COEFFICIENT, LOWER (ILE TO ITE) =
  .64128E+00   .20246E+00  -.62959E-01  -.33531E+00  -.51373E+00  -.56171E+00  -.62432E+00  -.64968E+00  -.68677E+00  -.79599E+00
```

```
                              *** 3-D CIRCULAR ARC ***

MACH NUMBER =   .72000E+00
SIMILARITY PARAMETER (K) =   .20807E+01
THICKNESS RATIO =   .90000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
WING ASPECT RATIO =   .20000E+01
CP SCALING FACTOR (CP/CPBAR) =   .14520E+00
CRITICAL PRESSURE COEFFICIENT (SONIC) =  -.60735E+00

AIRFOIL FORCE COEFFICIENTS

  LIFT =   .26288E-11
  MOMENT ABOUT (X=0) =  -.83099E-12

                  AIRFOIL SPANWISE COORDINATE =  0.          SECTION LIFT COEFFICIENT =   .50233E-11

 AIRFOIL STREAMWISE COORDINATE
  0.           .70000E-01   .14000E+00   .21000E+00   .28000E+00   .35000E+00   .42000E+00   .50000E+00   .55000E+00   .60000E+0
   .65000E+00   .70000E+00   .76000E+00   .82000E+00   .90000E+00   .10000E+01

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
   .72470E-01  -.20510E-01  -.83065E-01  -.13294E+00  -.17005E+00  -.19632E+00  -.21252E+00  -.21498E+00  -.21654E+00  -.20951E+0
  -.19763E+00  -.18098E+00  -.15370E+00  -.11029E+00  -.56796E-01   .58825E-01

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
   .72470E-01  -.20510E-01  -.83065E-01  -.13294E+00  -.17005E+00  -.19632E+00  -.21252E+00  -.21888E+00  -.21654E+00  -.20951E+0
  -.19763E+00  -.18098E+00  -.15370E+00  -.11029E+00  -.56796E-01   .58825E-01
```

```
*** 3-D CIRCULAR ARC ***

MACH NUMBER =    .80000E+00
SIMILARITY PARAMETER (K) =    .17044E+01
THICKNESS RATIO =    .60000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
WING ASPECT RATIO =    .20000E+01
CP SCALING FACTOR (CP/CPBAR) =    .13203E+00
CRITICAL PRESSURE COEFFICIENT (SONIC) =  -.40875E+00


AIRFOIL FORCE COEFFICIENTS

  LIFT =    .13146E-11
  MOMENT ABOUT (X=0) =  -.90707E-13

              AIRFOIL SPANWISE COORDINATE =  0.          SECTION LIFT COEFFICIENT =   .13827E-11

AIRFOIL STREAMWISE COORDINATE
  0.          .70000E-01   .14000E+00   .21000E+00   .28000E+00   .35000E+00   .42000E+00   .50000E+00   .55000E+00   .60000E+0
   .65000E+00   .70000E+00   .76000E+00   .82000E+00   .90000E+00   .10000E+01

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  .10071E+00  -.12127E-01  -.96758E-01  -.16573E+00  -.22032E+00  -.26095E+00  -.28722E+00  -.29914E+00  -.29654E+00  -.28591E+0
 -.26757E+00  -.24170E+00  -.20118E+00  -.15044E+00  -.86058E-01   .82138E-01

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  .10071E+00  -.12127E-01  -.96758E-01  -.16573E+00  -.22032E+00  -.26095E+00  -.28722E+00  -.29914E+00  -.29654E+00  -.28591E+0
 -.26757E+00  -.24170E+00  -.20118E+00  -.15044E+00  -.86058E-01   .82138E-01
```

```
***  3-D CIRCULAR ARC  ***

MACH NUMBER =   .84000E+00
SIMILARITY PARAMETER (K) =   .11500E+01
THICKNESS RATIO =   .60000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
WING ASPECT RATIO =   .20000E+01
CP SCALING FACTOR (CP/CPBAR) =   .12654E+00
CRITICAL PRESSURE COEFFICIENT (SONIC) =  -.29580E+00

AIRFOIL FORCE COEFFICIENTS

   LIFT =   .25957E-12
   MOMENT ABOUT (X=0) =   .72288E-13

            AIRFOIL SPANWISE COORDINATE =  0.          SECTION LIFT COEFFICIENT =   .21721E-12

AIRFOIL STREAMWISE COORDINATE
  0.          .70000E-01   .14000E+00   .21000E+00   .28000E+00   .35000E+00   .42000E+00   .50000E+00   .55000E+00   .60000E+00
  .65000E+00  .70000E+00   .74000E+00   .82000E+00   .99000E+00   .10000E+01

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  .16786E+00  .13317E-01  -.85958E-01  -.17192E+00  -.24734E+00  -.30989E+00  -.38627E+00  -.44596E+00  -.49143E+00  -.52335E+00
 -.50552E+00 -.46193E+00  -.20729E+00  -.86302E-01  -.10191E-01   .13845E+00

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  .16786E+00  .13317E-01  -.85958E-01  -.17192E+00  -.24736E+00  -.30849E+00  -.38627E+00  -.44596E+00  -.49143E+00  -.52335E+00
 -.50552E+00 -.46193E+00  -.20729E+00  -.86302E-01  -.10191E-01   .13805E+00
```

● FINAL OUTPUT OF RUN 4S

```
                              *** 3-D CIRCULAR ARC ***

MACH NUMBER =   .80000E+00
SIMILARITY PARAMETER (K) =   .11590E+01
THICKNESS RATIO =   .60000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
WING ASPECT RATIO =   .20000E+01
CP SCALING FACTOR (CP/CPBAR) =   .12658E+00
CRITICAL PRESSURE COEFFICIENT (SONIC) =  -.29340E+00

 AIRFOIL STREAMWISE COORDINATE
  0.           .70000E-01   .14000E+00   .21000E+00   .28000E+00   .35000E+00   .42000E+00   .50000E+00   .55000E+00   .60000E+0
   .65000E+00   .70000E+00   .76000E+00   .82000E+00   .90000E+00   .10000E+01

              AIRFOIL SPANWISE COORDINATE =  0.            SECTION LIFT COEFFICIENT =   .12637E-12

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
   .14741E+00   .13297E-01  -.85958E-01  -.17190E+00  -.24733E+00  -.30967E+00  -.38661E+00  -.46625E+00  -.49105E+00  -.52422E+0
  -.56667E+00  -.46052E+00  -.20463E+00  -.85482E-01  -.17553E-01   .13511E+00

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
   .14741E+00   .13297E-01  -.85958E-01  -.17190E+00  -.24733E+00  -.30967E+00  -.38661E+00  -.46625E+00  -.49105E+00  -.52422E+0
  -.56667E+00  -.46052E+00  -.20463E+00  -.85482E-01  -.17553E-01   .13511E+00

              AIRFOIL SPANWISE COORDINATE =   .25000E+00  SECTION LIFT COEFFICIENT =   .12772E-12

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
   .14742E+00   .13330E-01  -.85912E-01  -.17183E+00  -.24724E+00  -.30956E+00  -.38649E+00  -.46604E+00  -.49163E+00  -.52438E+0
  -.56569E+00  -.45623E+00  -.20197E+00  -.86381E-01  -.18002E-01   .13409E+00

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
   .14742E+00   .13330E-01  -.85912E-01  -.17183E+00  -.24724E+00  -.30956E+00  -.38649E+00  -.46604E+00  -.49163E+00  -.52438E+0
  -.56569E+00  -.45623E+00  -.20197E+00  -.86381E-01  -.18002E-01   .13409E+00

              AIRFOIL SPANWISE COORDINATE =   .50000E+00  SECTION LIFT COEFFICIENT =   .12547E-12

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
   .14746E+00   .13425E-01  -.85773E-01  -.17164E+00  -.24697E+00  -.30923E+00  -.38607E+00  -.46529E+00  -.49100E+00  -.52635E+0
  -.56137E+00  -.44062E+00  -.19445E+00  -.89398E-01  -.19488E-01   .13418E+00

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
   .14746E+00   .13425E-01  -.85773E-01  -.17164E+00  -.24697E+00  -.30923E+00  -.38607E+00  -.46529E+00  -.49100E+00  -.52635E+0
  -.56137E+00  -.44062E+00  -.19445E+00  -.89398E-01  -.19488E-01   .13418E+00

              AIRFOIL SPANWISE COORDINATE =   .75000E+00  SECTION LIFT COEFFICIENT =   .11962E-12

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
```

● FINAL OUTPUT OF RUN 4S (CONT'D)

```
 .14749E+00  .13575E-01 -.65519E-01 -.17124E+00 -.24467E+00 -.30459E+00 -.38523E+00 -.46356E+00 -.49072E+00 -.52387E+0
-.55234E+00 -.41949E+00 -.18133E+00 -.94211E-01 -.21944E-01  .13295E+00
AIRFOIL PRESSURE COEFFICIENTS, LOWER =
 .14749E+00  .13575E-01 -.65519E-01 -.17124E+00 -.24467E+00 -.30459E+00 -.38523E+00 -.46356E+00 -.49072E+00 -.52387E+0
-.55238E+00 -.41949E+00 -.18133E+00 -.94211E-01 -.21944E-01  .13295E+00

          AIRFOIL SPANWISE COORDINATE =   .10000E+01  SECTION LIFT COEFFICIENT =   .11018E-12

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 .14745E+00  .13744E-01 -.65210E-01 -.17075E+00 -.24541E+00 -.30751E+00 -.38355E+00 -.46047E+00 -.48735E+00 -.51633E+0
-.53454E+00 -.38471E+00 -.16334E+00 -.10403E+00 -.26594E-01  .13054E+00
AIRFOIL PRESSURE COEFFICIENTS, LOWER =
 .14745E+00  .13744E-01 -.65210E-01 -.17075E+00 -.24541E+00 -.30751E+00 -.38355E+00 -.46047E+00 -.48735E+00 -.51633E+0
-.53454E+00 -.38471E+00 -.16334E+00 -.10403E+00 -.26594E-01  .13054E+00

          AIRFOIL SPANWISE COORDINATE =   .12500E+01  SECTION LIFT COEFFICIENT =   .98934E-13

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 .14714E+00  .13034E-01 -.64810E-01 -.16989E+00 -.24498E+00 -.30553E+00 -.37976E+00 -.45275E+00 -.47645E+00 -.49805E+0
-.46715E+00 -.31702E+00 -.17012E+00 -.12594E+00 -.35915E-01  .12428E+00
AIRFOIL PRESSURE COEFFICIENTS, LOWER =
 .14714E+00  .13034E-01 -.64810E-01 -.16989E+00 -.24498E+00 -.30553E+00 -.37976E+00 -.45275E+00 -.47645E+00 -.49805E+0
-.46715E+00 -.31702E+00 -.17012E+00 -.12594E+00 -.35915E-01  .12428E+00

          AIRFOIL SPANWISE COORDINATE =   .15000E+01  SECTION LIFT COEFFICIENT =   .84990E-13

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 .14549E+00  .13449E-01 -.64427E-01 -.16829E+00 -.24079E+00 -.30049E+00 -.36924E+00 -.43067E+00 -.44702E+00 -.45550E+0
-.39274E+00 -.26511E+00 -.14997E+00 -.13631E+00 -.41347E-01  .12240E+00
AIRFOIL PRESSURE COEFFICIENTS, LOWER =
 .14549E+00  .13449E-01 -.64427E-01 -.16829E+00 -.24079E+00 -.30049E+00 -.36924E+00 -.43067E+00 -.44702E+00 -.45550E+0
-.39274E+00 -.26511E+00 -.14997E+00 -.13631E+00 -.41347E-01  .12240E+00

          AIRFOIL SPANWISE COORDINATE =   .17500E+01  SECTION LIFT COEFFICIENT =   .66550E-13

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 .14088E+00  .10997E-01 -.64138E-01 -.16370E+00 -.23032E+00 -.28574E+00 -.33330E+00 -.37723E+00 -.38674E+00 -.35442E+0
-.30158E+00 -.25492E+00 -.20229E+00 -.14211E+00 -.46583E-01  .11543E+00
AIRFOIL PRESSURE COEFFICIENTS, LOWER =
 .14088E+00  .10997E-01 -.64138E-01 -.16370E+00 -.23032E+00 -.28574E+00 -.33330E+00 -.37723E+00 -.38674E+00 -.35442E+0
-.30158E+00 -.25492E+00 -.20229E+00 -.14211E+00 -.46583E-01  .11543E+00

          AIRFOIL SPANWISE COORDINATE =   .19000E+01  SECTION LIFT COEFFICIENT =   .49467E-13

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 .12844E+00  .49354E-02 -.82564E-01 -.15230E+00 -.20984E+00 -.25241E+00 -.28066E+00 -.30976E+00 -.30885E+00 -.29051E+0
-.26203E+00 -.23022E+00 -.18655E+00 -.13490E+00 -.49378E-01  .10280E+00
AIRFOIL PRESSURE COEFFICIENTS, LOWER =
 .12844E+00  .49354E-02 -.82564E-01 -.15230E+00 -.20984E+00 -.25241E+00 -.28066E+00 -.30976E+00 -.30885E+00 -.29051E+0
-.26203E+00 -.23022E+00 -.18655E+00 -.13490E+00 -.49378E-01  .10280E+00
```

● FINAL OUTPUT OF RUN 4S (CONT'D)

```
AIRFOIL SPANWISE COORDINATE =   .20000E+01  SECTION LIFT COEFFICIENT =   .32802E-13

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  .10341E+00  -.30669E-02  -.74786E-01  -.12979E+00  -.17318E+00  -.20706E+00  -.23165E+00  -.24531E+00  -.24325E+00  -.23176E+0
 -.21325E+00  -.19005E+00  -.15657E+00  -.11611E+00  -.47778E-01   .80783E-01

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  .10341E+00  -.30669E-02  -.74786E-01  -.12979E+00  -.17318E+00  -.20706E+00  -.23165E+00  -.24531E+00  -.24325E+00  -.23176E+0
 -.21325E+00  -.19005E+00  -.15657E+00  -.11611E+00  -.47778E-01   .80783E-01
```

RUN 4S

```
***  3-D CIRCULAR ARC  ***

MACH NUMBER =   .80000E+00
SIMILARITY PARAMETER (K) =   .11500E+01
THICKNESS RATIO =   .60000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
WING ASPECT RATIO =   .20000E+01
CP SCALING FACTOR (CP/CPHAT) =   .12458E+00
CRITICAL PRESSURE COEFFICIENT (SONIC) =  -.29340E+00

AIRFOIL FORCE COEFFICIENTS

 LIFT =   .17274E-12
 MOMENT ABOUT (X=0) =   .67275E-13

              AIRFOIL SPANWISE COORDINATE =  0.              SECTION LIFT COEFFICIENT =   .12437E-12

AIRFOIL STREAMWISE COORDINATE
  0.            .70000E-01   .14000E+00   .21000E+00   .28000E+00   .35000E+00   .42000E+00   .50000E+00   .55000E+00   .60000E+0
  .65000E+00    .70000E+00   .76000E+00   .82000E+00   .90000E+00   .10000E+01

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  .10701E+00    .13297E-01  -.65958E-01  -.17190E+00  -.26733E+00  -.30967E+00  -.39601E+00  -.46625E+00  -.49105E+00  -.52422E+0
 -.54667E+00   -.46052E+00  -.20063E+00  -.95802E-01  -.17553E-01   .13511E+00

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  .10701E+00    .13297E-01  -.65958E-01  -.17190E+00  -.26733E+00  -.30967E+00  -.39601E+00  -.46625E+00  -.49105E+00  -.52422E+0
 -.54667E+00   -.46052E+00  -.20063E+00  -.85802E-01  -.17553E-01   .13511E+00
```

7.2 TDUTRN Test Cases

A sequence of TDUTRN runs are described in this section which calculate the unsteady flow perturbation for the previously described circular arc rectangular wing oscillating in pitch about the leading edge at:

$$M_\infty = .86 \quad \left| \begin{array}{l} k = 0.0 \\ \\ k = 0.1 \end{array} \right.$$

The individual runs required to calculate these cases are described in the run log in Table 2. All runs used the steady solution given on tape dump 4S. The table presents the reduced frequency, restart tape read, grid iterations, convergence achieved and tape dump generated. The runs were calculated using the same grid as the steady runs. They were performed in the order shown to implement the "bootstrapping" technique for getting from one reduced frequency to another. The input required for each run and sample output are presented in the following section.

Run	k	Restart Tape Used	Grid Iteration	Convergence Achieved	Tape Dump Generated
1U	0.0	--	50	2.2×10^{-3}	1U
2U	0.0	1U	100	1.9×10^{-4}	2U
3U	0.0	2U	50	6.8×10^{-5}	3U
4U*	0.0	3U	86	1.0×10^{-5}	4U
5U	0.05	4U	50	1.3×10^{-3}	5U
6U*	0.1	5U	200	6.0×10^{-5}	6U

TABLE 2. SEQUENCE OF RUNS FOR TDUTRN SAMPLE CASES

(*DENOTES CONVERGED SOLUTION)

7.2.1 Input for TDUTRN Test Cases

The card input for each of the TDUTRN runs described above is given in this section.

- Run 1U: read file 4S, no restart tape read; generate file 1U.

```
                    ***3D CIRCULAR ARC***

$CØNTRL
ITAPE=0,
$END
$IN
X(1)=-3.2,-2.2,-1.5,-1.02,-.67,-.42,-.24,-.1,0.,.07,
     .14,.21,.28,.35,.42,.5,.55,.6,.65,.7,.76,.82,
     .9,1.,1.14,1.34,1.62,2.02,2.58,3.38,

Y(1)=-5.4,-3.41,-2.91,-1.91,-1.21,-.74,-.43,-.22,-.08,
     0.,.08,.22,.43,.74,1.21,1.91,2.91,3.41,4.3,

Z(1)=0.,.25,.5,.75,1.,1.25,1.5,1.75,1.9,2.,2.1,2.25,
     2.45,2.75,3.2,3.85,4.75,6.,6.8,
IM=30,
JM=19,
KM=19,
ILE=9,
ITE=24,
JW=10,
KSPAN=10,
GAMFF(1)=10*(1.,0.),
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSGRD(1)=1.E-4,
NDUMP=2000,
NGRID=50,
NGFF=10,
PGFF=1.5,
KEPS=1,
NPRINT=5,
NNPRT=10,
KPRT(1)=1,2,3,4,5,6,7,8,9,10,
SMALLK=0.0,
IK=0,
XP=0.0,
ITYPE=1,
IØPT=0,
ZE(1)=0.,.125,.25,.375,.5,.625,.75,.875,1.,1.125,1.25,
                1.375,1.5,1.625,1.75,1.825,1.9,1.95,2.,
NZE=19,
$END
```

- Run 2U: read file 4S, restart file 1U; generate file 2U

3D CIRCULAR ARC

```
$CØNTRL
ITAPE=1,
$END
$IN
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSGRD=1.E-4,
NDUMP=2000,
NGRID=100,
NGFF=10,
PGFF=1.5,
KEPS=1,
NPRINT=5,
NKPRT=10,
KPRT=1,2,3,4,5,6,7,8,9,10,
ITYPE=1,
IØPT=0,
ZE(1)=0.,.125,.25,.375,.5,.625,.75,.875,1.,1.125,1.25,
      1.375,1.5,1.625,1.75,1.825,1.9,1.95,2.,
NZE=19,
IK=0,
$END
```

- Run 3U: read file 4S, restart file 2U; generate file 3U

3D CIRCULAR ARC

```
$CØNTRL
ITAPE=1,
$END
$IN
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSGRD=1.E-4,
NDUMP=2000,
NGRID=50,
NGFF=10,
PGFF=1.5,
KEPS=1,
NPRINT=5,
NKPRT=10,
KPRT=1,2,3,4,5,6,7,8,9,10,
```

```
ITYPE=1,
IØPT=0,
ZE(1)=0.,.25,.25,.375,.5,.625,.75,.875,1.,1.125,1.25,
      1.375,1.5,1.625,1.75,1.825,1.9,1.95,2.,
NZE=19,
IK= 0,
$END
```

- Run 4U: read file 4S, restart file 3U; generate file 4U

```
                ***3D CIRCULAR ARC***

$CØNTRL
ITAPE=1,
$END
$IN
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSGRD=1.E-5,
NDUMP=2000,
NGRID=100,
NGFF=10,
PGFF=1.5,
KEPS=1,
NPRINT=5,
NKPRT=10,
KPRT=1,2,3,4,5,6,7,8,9,10,
ITYPE=1,
IØPT=0,
ZE(1)=0.,.125,.25,.375,.5,.625,.75,.875,1.,1.125,1.25,
      1.375,1.5,1.625,1.75,1.825,1.9,1.95,2.,
NZE=19,
IK=0,
$END
```

- Run 5U: read file 4S, restart file 4U; generate file 5U

```
                ***3D CIRCULAR ARC***

$CØNTRL
ITAPE=1,
$END
$IN
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSGRD=1.E-5,
NDUMP=2000,
```

```
NGRID=50,
NGFF=10,
PGFF=1.5,
KEPS=1,
NPRINT=10,
NKPRT=10,
KPRT=1,2,3,4,5,6,7,8,9,10,
ITYPE=1,
IØPT=0,
ZE(1)=0.,.125,.25,.375,.5,.625,.75,.875,1.,1.125,1.25,
      1.375,1.5,1.625,1.75,1.825,1.9,1.95,2.,
NZE=19,
IK=1,
SMALLK=.05,
$END
```

- Run 6U: read file 4S, restart file 5U; generate file 6U

```
                ***3D CIRCULAR ARC***

$CØNTRL,
ITAPE=1,
$END
$IN
ØMEGAH=.75,
ØMEGAE=1.7,
ØMEGAP=.75,
EPSGRD=1.E-5,
NDUMP=2000,
NGRID=200,
NGFF=10,
PGFF=1.5,
KEPS=1,
NPRINT=10,
NKPRT=10,
KPRT=1,2,3,4,5,6,7,8,9,10,
ITYPE=1,
IØPT=0,
ZE(1)=0.,.125,.25,.375,.5,.625,.75,.875,1.,1.125,1.25,
      1.375,1.5,1.625,1.75,1.825,1.9,1.95,2.,
NZE=19,
IK=1,
SMALLK=.1,
$END
```

7.2.2 Sample Output for TDUTRN Test Cases

The following pages contain a sample of the continuous commentary output for the first 9 cycles of Run 1U in addition to the final printed page for all runs. Also included in the complete final output for the final run.

● SAMPLE OUTPUT FOR RUN 1U

```
SIMILARITY PARAMETER (K) =   .11590E+01
SCALING FACTOR (CP/CPMAR) =   .12658E+00

AT ITERATION    1 THE MAXIMUM ERROR =  -.74325E+00 -0.          AND OCCURRED AT NODE  144

AT ITERATION    2 THE MAXIMUM ERROR =   .48110E+00  0.          AND OCCURRED AT NODE  976

AT ITERATION    3 THE MAXIMUM ERROR =  -.30637E+00 -0.          AND OCCURRED AT NODE  391

AT ITERATION    4 THE MAXIMUM ERROR =   .69670E+00  0.          AND OCCURRED AT NODE 1222

AT ITERATION    4 AND K =  1 SCALED PRESSURE COEFFICIENT, UPPER (ILE TO ITE) =
 -.41573E+01 -0.          -.25934E+01 -0.          -.11132E+01 -0.           .35641E+00 -0.          -.21296E+01 -0.
 -.30578E+01 -0.          -.22230E+01 -0.          -.80147E+00 -0.          -.72387E+00 -0.          -.25205E+01 -0.
 -.57412E+01 -0.           .11056E+00 -0.           .62177E+01 -0.           .89903E+00 -0.           .36998E+00 -0.
  .87403E+00 -0.

AT ITERATION    4 AND K =  1 SCALED PRESSURE COEFFICIENT, LOWER (ILE TO ITE) =
  .41573E+01 -0.           .25934E+01 -0.           .11132E+01 -0.          -.35641E+00 -0.           .21296E+01 -0.
  .30578E+01 -0.           .22230E+01 -0.           .80147E+00 -0.           .72387E+00 -0.           .25205E+01 -0.
  .57412E+01 -0.          -.11056E+00 -0.          -.62177E+01 -0.          -.89903E+00 -0.          -.36998E+00 -0.
 -.47403E+00 -0.

AT ITERATION    4 AND K = 10 SCALED PRESSURE COEFFICIENT, UPPER (ILE TO ITE) =
 -.12657E+01 -0.          -.86581E+00 -0.          -.63181E+00 -0.          -.37856E+00 -0.          -.47659E+00 -0.
 -.35078E+00 -0.           .29746E-02 -0.          -.22669E-01 -0.           .39926E-01 -0.           .51696E-02 -0.
 -.33621E+01 -0.          -.28481E+01 -0.           .12504E+01 -0.          -.12096E-01 -0.          -.11674E-03 -0.
 -.10736E+00 -0.

AT ITERATION    4 AND K = 10 SCALED PRESSURE COEFFICIENT, LOWER (ILE TO ITE) =
  .12657E+01 -0.           .86581E+00 -0.           .63181E+00 -0.           .37856E+00 -0.           .47659E+00 -0.
  .35078E+00 -0.          -.29746E-02 -0.           .22669E-01 -0.          -.39926E-01 -0.          -.51696E-02 -0.
  .33621E+01 -0.           .28481E+01 -0.          -.12504E+01 -0.           .12096E-01 -0.           .11674E-03 -0.
  .10736E+00 -0.

AT ITERATION    5 THE MAXIMUM ERROR =  -.31044E+00 -0.          AND OCCURRED AT NODE  924

AT ITERATION    6 THE MAXIMUM ERROR =   .31816E+00  0.          AND OCCURRED AT NODE  924

AT ITERATION    7 THE MAXIMUM ERROR =   .14791E+00  0.          AND OCCURRED AT NODE  881

AT ITERATION    8 THE MAXIMUM ERROR =   .13344E+00  0.          AND OCCURRED AT NODE  862

AT ITERATION    9 THE MAXIMUM ERROR =   .88338E-01  0.          AND OCCURRED AT NODE  843

AT ITERATION    9 AND K =  1 SCALED PRESSURE COEFFICIENT, UPPER (ILE TO ITE) =
 -.33233E+01 -0.          -.20994E+01 -0.          -.18761E+01 -0.          -.21180E+01 -0.          -.19763E+01 -0.
 -.16974E+01 -0.          -.15051E+01 -0.          -.86306E+00 -0.          -.10884E+01 -0.          -.13020E+01 -0.
 -.19131E+01 -0.          -.17273E+01 -0.          -.17289E+00 -0.           .22416E+00 -0.           .22061E+00 -0.
  .30828E+00 -0.

AT ITERATION    9 AND K =  1 SCALED PRESSURE COEFFICIENT, LOWER (ILE TO ITE) =
  .33233E+01 -0.           .20994E+01 -0.           .18761E+01 -0.           .21180E+01 -0.           .19763E+01 -0.
  .16974E+01 -0.           .15051E+01 -0.           .86306E+00 -0.           .10884E+01 -0.           .13020E+01 -0.
  .19131E+01 -0.           .17273E+01 -0.           .17289E+00 -0.          -.22416E+00 -0.          -.22061E+00 -0.
 -.30828E+00 -0.
```

RUN 1U

```
***  3-D CIRCULAR ARC  ***

MACH NUMBER =   .94000E+00
SIMILARITY PARAMETER =   .11590E+01
THICKNESS RATIO =   .60000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
REDUCED FREQUENCY (BASED ON CHORD) =  0.
SCALED FREQUENCY (OMEGA) =  0.
PITCH AXIS (XP) =  0.
WING ASPECT RATIO =   .20000E+01
CP SCALING FACTOR (CP/CPBAR) =   .12050E+00

UNSTEADY FORCE COEFFICIENTS (PER UNIT PITCH ANGLE IN RADIANS)

  LIFT =   .13802E+02  0.
  MOMENT ABOUT (XREF) =  0.          0.

        AIRFOIL SPANWISE COORDINATE =  0.         SECTION LIFT COEFFICIENT =   .97814E+01  0.

PRESSURE COEFFICIENTS ON THE AIRFOIL (PER UNIT PITCH ANGLE IN RADIANS)

 AIRFOIL STREAMWISE COORDINATE
  0.                  .70000E-01           .14000E+00           .21000E+00           .28000E+00
  .35000E+00          .42000E+00           .50000E+00           .55000E+00           .60000E+00
  .65000E+00          .70000E+00           .76000E+00           .82000E+00           .90000E+00
  .10000E+01

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  -.72729E+01 -0.     -.51714E+01 -0.      -.47837E+01 -0.      -.44924E+01 -0.      -.43506E+01 -0.
  -.40981E+01 -0.     -.29819E+01 -0.      -.29881E+01 -0.      -.36883E+01 -0.      -.27724E+01 -0.
  -.52932E+01 -0.     -.98460E+01 -0.      -.62583E+01 -0.      -.13594E+01 -0.      -.82691E+00 -0.
  -.39257E+00 -0.

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  .72729E+01 -0.      .51714E+01 -0.       .47837E+01 -0.       .44924E+01 -0.       .43506E+01 -0.
  .40981E+01 -0.      .29819E+01 -0.       .29881E+01 -0.       .36883E+01 -0.       .27724E+01 -0.
  .52932E+01 -0.      .98460E+01 -0.       .62583E+01 -0.       .13594E+01 -0.       .82691E+00 -0.
  .39257E+00 -0.
```

```
*** 3-D CIRCULAR ARC ***

MACH NUMBER =   .80000E+00
SIMILARITY PARAMETER =   .11590E+01
THICKNESS RATIO =   .80000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
REDUCED FREQUENCY (BASED ON CHORD) =  0.
SCALED FREQUENCY (OMEGA) =  0.
PITCH AXIS (XP) =  0.
WING ASPECT RATIO =   .20000E+01
CP SCALING FACTOR (CP/CPHAR) =   .12458E+00

UNSTEADY FORCE COEFFICIENTS (PER UNIT PITCH ANGLE IN RADIANS)

  LIFT =   .16042E+02  0.
  MOMENT ABOUT (XP) =  0.             0.

           AIRFOIL SPANWISE COORDINATE =  0.          SECTION LIFT COEFFICIENT =   .10259E+02  0.

PRESSURE COEFFICIENTS ON THE AIRFOIL (PER UNIT PITCH ANGLE IN RADIANS)

 AIRFOIL STREAMWISE COORDINATE
  0.                 .70000E-01         .14000E+00         .21000E+00         .28000E+00
   .35000E+00        .42000E+00         .59000E+00         .55000E+00         .60000E+00
   .65000E+00        .70000E+00         .76000E+00         .82000E+00         .90000E+00
   .10000E+01

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  -.74684E+01 -0.    -.54481E+01 -0.    -.50367E+01 -0.    -.47265E+01 -0.    -.45700E+01 -0.
  -.42709E+01 -0.    -.32196E+01 -0.    -.31087E+01 -0.    -.39608E+01 -0.    -.30991E+01 -0.
  -.54314E+01 -0.    -.10301E+02 -0.    -.62825E+01 -0.    -.13294E+01 -0.    -.81044E+00 -0.
  -.20592E+00 -0.

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
   .74684E+01 -0.     .54481E+01 -0.     .50367E+01 -0.     .47265E+01 -0.     .45760E+01 -0.
   .42709E+01 -0.     .32196E+01 -0.     .31487E+01 -0.     .39608E+01 -0.     .30991E+01 -0.
   .54314E+01 -0.     .10301E+02 -0.     .62825E+01 -0.     .13294E+01 -0.     .81044E+00 -0.
   .20592E+00 -0.
```

RUN 2U

*** 3-D CIRCULAR ARC ***

MACH NUMBER = .84000E+00
SIMILARITY PARAMETER = .11540E+01
THICKNESS RATIO = .50000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) = 0.
REDUCED FREQUENCY (BASED ON C-ROOT) = 0.
SCALED FREQUENCY (OMEGA) = 0.
PITCH AXIS (XP) = 0.
WING ASPECT RATIO = .20000E+01
CP SCALING FACTOR (CP/CPBAR) = .12658E+00

UNSTEADY FORCE COEFFICIENTS (PER UNIT PITCH ANGLE IN RADIANS)

LIFT = .14088E+02 0.
MOMENT ABOUT (XBXP) = .55738E+01 0.

AIRFOIL SPANWISE COORDINATE = 0.   SECTION LIFT COEFFICIENT = .10293E+02 0.

PRESSURE COEFFICIENTS ON THE AIRFOIL (PER UNIT PITCH ANGLE IN RADIANS)

AIRFOIL STREAMWISE COORDINATE

0.	.70000E-01	.14000E+00	.21000E+00	.28000E+00
.35000E+00	.42000E+00	.50000E+00	.55000E+00	.63000E+00
.65000E+00	.70000E+00	.76000E+00	.82000E+00	.90000E+00
.10000E+01				

AIRFOIL PRESSURE COEFFICIENTS, UPPER =

-.76951E+01 -0.	-.54608E+01 -0.	-.50537E+01 -0.	-.47422E+01 -0.	-.45913E+01 -0.
-.42869E+01 -0.	-.32344E+01 -0.	-.32016E+01 -0.	-.39865E+01 -0.	-.31183E+01 -0.
-.58081E+01 -0.	-.10333E+02 -0.	-.62893E+01 -0.	-.13291E+01 -0.	-.81049E+00 -0.
-.20543E+00 -0.				

AIRFOIL PRESSURE COEFFICIENTS, LOWER =

.76951E+01 -0.	.54608E+01 -0.	.50537E+01 -0.	.47422E+01 -0.	.45913E+01 -0.
.42869E+01 -0.	.32344E+01 -0.	.32016E+01 -0.	.39865E+01 -0.	.31193E+01 -0.
.58081E+01 -0.	.10333E+02 -0.	.62893E+01 -0.	.13291E+01 -0.	.81049E+00 -0.
.20543E+00 -0.				

```
*** 3-D CIRCULAR ARC ***

MACH NUMBER =   .80000E+00
SIMILARITY PARAMETER =   .11590E+01
THICKNESS RATIO =   .40000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
REDUCED FREQUENCY (BASED ON CHORD) =  0.
SCALED FREQUENCY (OMEGA) =  0.
PITCH AXIS (XP) =  0.
WING ASPECT RATIO =   .20000E+01
CP SCALING FACTOR (CP/CPBAR) =   .12650E+00

UNSTEADY FORCE COEFFICIENTS (PER UNIT PITCH ANGLE IN RADIANS)

  LIFT =   .14104E+02  0.
  MOMENT ABOUT (X=XP) =   .55805E+01  0.

        AIRFOIL SPANWISE COORDINATE =  0.          SECTION LIFT COEFFICIENT =   .10300E+02  0.

PRESSURE COEFFICIENTS ON THE AIRFOIL (PER UNIT PITCH ANGLE IN RADIANS)

  AIRFOIL STREAMWISE COORDINATE
   0.                  .70000E-01          .14000E+00          .21000E+00          .28000E+00
    .35000E+00         .42000E+00          .50000E+00          .55000E+00          .60000E+00
    .65000E+00         .70000E+00          .76000E+00          .82000E+00          .90000E+00
    .10000E+01

  AIRFOIL PRESSURE COEFFICIENTS, UPPER =
   -.77059E+01 -0.     -.54741E+01 -0.     -.50606E+01 -0.     -.47407E+01 -0.     -.45975E+01 -0.
   -.42930E+01 -0.     -.32404E+01 -0.     -.32069E+01 -0.     -.39937E+01 -0.     -.31262E+01 -0.
   -.50772E+01 -0.     -.10347E+02 -0.     -.62921E+01 -0.     -.13291E+01 -0.     -.01050E+00 -0.
   -.20523E+00 -0.

  AIRFOIL PRESSURE COEFFICIENTS, LOWER =
    .77059E+01 -0.      .54741E+01 -0.      .50606E+01 -0.      .47407E+01 -0.      .45975E+01 -0.
    .42930E+01 -0.      .32404E+01 -0.      .32069E+01 -0.      .39937E+01 -0.      .31262E+01 -0.
    .50772E+01 -0.      .10347E+02 -0.      .62921E+01 -0.      .13291E+01 -0.      .01050E+00 -0.
    .20523E+00 -0.
```

RUN 4U

```
*** 3-D CIRCULAR ARC ***

MACH NUMBER =   .80000E+00
SIMILARITY PARAMETER =   .11590E+01
THICKNESS RATIO =   .60000E-01
AIRFOIL ANGLE OF ATTACK (RADIANS) =  0.
REDUCED FREQUENCY (BASED ON CHORD) =   .50000E-01
SCALED FREQUENCY (OMEGA) =   .10459E+00
PITCH AXIS (XP) =  0.
WING ASPECT RATIO =   .20000E+01
CP SCALING FACTOR (CP/CPBAR) =   .12650E+00

UNSTEADY FORCE COEFFICIENTS (PER UNIT PITCH ANGLE IN RADIANS)

  LIFT =   .13059E+02  -.10000E+01
  MOMENT ABOUT (X=XP) =   .55060E+01  -.82299E+00

        AIRFOIL SPANWISE COORDINATE =  0.          SECTION LIFT COEFFICIENT =   .10105E+02  -.12334E+01

PRESSURE COEFFICIENTS ON THE AIRFOIL (PER UNIT PITCH ANGLE IN RADIANS)

  AIRFOIL STREAMWISE COORDINATE
   0.                  .70000E-01          .14000E+00          .21000E+00          .28000E+00
    .35000E+00         .42000E+00          .50000E+00          .55000E+00          .60000E+00
    .65000E+00         .70000E+00          .76000E+00          .82000E+00          .90000E+00
    .10000E+01

  AIRFOIL PRESSURE COEFFICIENTS, UPPER =
   -.75049E+01   .11334E+01  -.53341E+01   .70093E+00  -.49310E+01   .73005E+00  -.46274E+01   .49716E+00  -.44817E+01   .47720E+0
   -.81937E+01   .61799E+00  -.31754E+01   .43296E+00  -.31400E+01   .41134E+00  -.39212E+01   .52100E+00  -.30016E+01   .47297E+0
   -.57703E+01   .80012E+00  -.10250E+02   .83145E+00  -.62799E+01   .01093E-01  -.13259E+01  -.56438E-01  -.00977E+00  -.43290E-0
   -.20512E+00  -.10963E-01

  AIRFOIL PRESSURE COEFFICIENTS, LOWER =
    .75049E+01  -.11334E+01   .53341E+01  -.70093E+00   .49310E+01  -.73005E+00   .46274E+01  -.49716E+00   .44817E+01  -.47720E+0
    .81937E+01  -.61790E+00   .31750E+01  -.43296E+00   .31400E+01  -.41130E+00   .39212E+01  -.52100E+00   .30016E+01  -.47297E+0
    .57703E+01  -.80012E+00   .10250E+02  -.83145E+00   .62799E+01  -.01093E-01   .13259E+01   .56438E-01   .00977E+00   .43290E-0
    .20512E+00   .10963E-01
```

● FINAL OUTPUT FOR RUN 6U

```
*** 3-D CIRCULAR ARC ***

MACH NUMBER = 8.00000E-01
SIMILARITY PARAMETER = 1.15096E+00
THICKNESS RATIO = 6.00000E-02
AIRFOIL ANGLE OF ATTACK (RADIANS) = 0.
REDUCED FREQUENCY (BASED ON CHORD) = 1.00000E-01
SCALED FREQUENCY (OMEGA) = 3.29173E-01
PITCH AXIS (XP) = 0.
WING ASPECT RATIO = 2.00000E+00
CP SCALING FACTOR (CP/CPBAR) = 1.26500E-01

 AIRFOIL STREAMWISE COORDINATE
   0.                  7.00000E-02         1.40000E-01         2.10000E-01         2.80000E-01
   3.50000E-01         4.20000E-01         5.00000E-01         5.50000E-01         6.00000E-01
   6.50000E-01         7.00000E-01         7.60000E-01         8.20000E-01         9.00000E-01
   1.00000E+00

        AIRFOIL SPANWISE COORDINATE = 0.          SECTION LIFT COEFFICIENT = 9.19540E+00 -2.50649E+00

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  -6.63692E+00  2.25370E+00 -4.73187E+00  1.57457E+00 -9.37073E+00  1.07132E+00 -9.11074E+00  1.39650E+00 -3.90762E+00  1.36229E+0
  -3.73633E+00  1.28218E+00 -2.84761E+00  9.37419E-01 -2.05523E+00  8.80020E-01 -3.55076E+00  1.13977E+00 -2.69540E+00  1.05751E+0
  -5.19909E+00  1.98264E+00 -9.79159E+00  1.96619E+00 -6.27650E+00  3.09413E-01 -1.33110E+00 -9.72575E-02 -8.06302E-01 -7.04077E-0
  -2.82616E-01 -3.02769E-02

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
   6.63692E+00 -2.25370E+00  4.73187E+00 -1.57457E+00  9.37073E+00 -1.07132E+00  9.11074E+00 -1.39650E+00  3.90762E+00 -1.36229E+0
   3.73633E+00 -1.28218E+00  2.84761E+00 -9.37419E-01  2.05523E+00 -8.80020E-01  3.55076E+00 -1.13977E+00  2.69540E+00 -1.05751E+0
   5.19909E+00 -1.98264E+00  9.79159E+00 -1.96619E+00  6.27650E+00 -3.09413E-01  1.33110E+00  9.72575E-02  8.06302E-01  7.04077E-0
   2.82616E-01  3.02769E-02

        AIRFOIL SPANWISE COORDINATE = 2.50000E-01  SECTION LIFT COEFFICIENT = 9.14278E+00 -2.50797E+00

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  -6.62691E+00  2.20129E+00 -4.72371E+00  1.56554E+00 -9.36910E+00  1.06272E+00 -9.10309E+00  1.38017E+00 -3.97927E+00  1.35388E+0
  -3.72760E+00  1.27012E+00 -2.08044E+00  9.31607E-01 -2.04098E+00  8.08197E-01 -3.45041E+00  1.10749E+00 -2.99787E+00  1.10183E+0
  -5.61217E+00  2.11553E+00 -9.46031E+00  1.87103E+00 -5.90489E+00  1.73065E-01 -1.38036E+00 -1.13054E-01 -8.10335E-01 -8.07056E-0
  -2.01793E-01 -3.30955E-02

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
   6.62691E+00 -2.20129E+00  4.72371E+00 -1.56554E+00  9.36910E+00 -1.06272E+00  9.10309E+00 -1.38017E+00  3.97927E+00 -1.35388E+0
   3.72760E+00 -1.27012E+00  2.08044E+00 -9.31607E-01  2.04098E+00 -8.08197E-01  3.45041E+00 -1.10749E+00  2.99787E+00 -1.10183E+0
   5.61217E+00 -2.11553E+00  9.46031E+00 -1.87103E+00  5.90489E+00 -1.73065E-01  1.38036E+00  1.13054E-01  8.10335E-01  8.07056E-0
   2.01793E-01  3.30955E-02

        AIRFOIL SPANWISE COORDINATE = 5.00000E-01  SECTION LIFT COEFFICIENT = 9.08346E+00 -2.50730E+00
```

● FINAL OUTPUT FOR RUN 6U (CONT'D)

```
AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 -6.59663E+00  2.20309E+00 -4.40047E+00  1.53773E+00 -6.34660E+00  1.03015E+00 -6.07845E+00  1.36234E+00 -3.95220E+00  1.32776E+0
 -3.69930E+00  1.24910E+00 -2.81781E+00  9.18278E-01 -2.02075E+00  4.71120E-01 -2.35140E+00  7.00007E-01 -6.33677E+00  1.72615E+0
 -8.09773E+00  3.00760E+00 -8.29632E+00  1.47685E+00 -4.53370E+00 -3.48090E-01 -1.32203E+00 -2.01580E-01 -7.72790E-01 -1.22814E-0
 -2.05960E-01 -4.72003E-02

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  6.59663E+00 -2.20309E+00  4.40047E+00 -1.53773E+00  6.34660E+00 -1.03015E+00  6.07845E+00 -1.36234E+00  3.95220E+00 -1.32776E+0
  3.69930E+00 -1.24910E+00  2.81781E+00 -9.18278E-01  2.02075E+00 -4.71120E-01  2.35140E+00 -7.00007E-01  6.33677E+00 -1.72615E+0
  8.09773E+00 -3.00760E+00  8.29632E+00 -1.47685E+00  8.53370E+00  3.48090E-01  1.32203E+00  2.01580E-01  7.72790E-01  1.22814E-0
  2.05500E-01  4.72003E-02

         AIRFOIL SPANWISE COORDINATE =  7.50000E-01  SECTION LIFT COEFFICIENT =  8.87913E+00 -2.39300E+00

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 -6.53359E+00  2.13585E+00 -6.65180E+00  1.08860E+00 -4.24780E+00  1.30910E+00 -6.03115E+00  1.31007E+00 -3.40000E+00  1.28122E+0
 -3.04529E+00  1.20687E+00 -2.78180E+00  8.86200E-01 -2.79805E+00  7.96083E-01 -2.07333E+00  1.01017E+00 -4.54915E+00  1.82219E+0
 -8.10711E+00  2.89247E+00 -7.66297E+00  1.27668E+00 -3.70303E+00 -6.09701E-01 -1.39035E+00 -3.00000E-01 -7.56303E-01 -1.71652E-0
 -2.49747E-01 -6.29000E-02

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  6.53359E+00 -2.13585E+00  6.65180E+00 -1.08860E+00  4.24780E+00 -1.30910E+00  6.03115E+00 -1.31007E+00  3.40000E+00 -1.28122E+0
  3.04529E+00 -1.20687E+00  2.78180E+00 -8.86200E-01  2.79805E+00 -7.96083E-01  2.07333E+00 -1.01017E+00  4.54915E+00 -1.82239E+0
  8.10711E+00 -2.89247E+00  7.64297E+00 -1.27668E+00  3.70303E+00  6.09701E-01  1.39035E+00  3.00000E-01  7.56303E-01  1.71452E-0
  2.49797E-01  6.29000E-02

         AIRFOIL SPANWISE COORDINATE =  1.00000E+00  SECTION LIFT COEFFICIENT =  8.60250E+00 -2.20760E+00

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 -6.02909E+00  2.03395E+00 -4.57173E+00  1.41023E+00 -4.21779E+00  1.31705E+00 -3.96951E+00  1.24679E+00 -3.00800E+00  1.20960E+0
 -3.55093E+00  1.13606E+00 -2.72030E+00  8.00600E-01 -2.09677E+00  8.20910E-01 -3.39173E+00  1.20060E+00 -5.19823E+00  1.95078E+0
 -8.06039E+00  2.50898E+00 -7.32185E+00  9.57906E-01 -2.35675E+00 -8.84906E-01 -1.03016E+00 -6.37270E-01 -4.93262E-01 -2.15430E-0
 -1.08556E-01 -7.20681E-02

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  6.02909E+00 -2.03345E+00  6.57173E+00 -1.01023E+00  4.21779E+00 -1.31745E+00  3.96951E+00 -1.24079E+00  3.00080E+00 -1.20960E+0
  3.55093E+00 -1.13606E+00  2.72030E+00 -8.00600E-01  2.04677E+00 -8.20910E-01  3.39123E+00 -1.20060E+00  5.19823E+00 -1.95078E+0
  8.06030E+00 -2.50898E+00  7.32105E+00 -9.57906E-01  2.35075E+00  8.80906E-01  1.03016E+00  4.37270E-01  4.93262E-01  2.15634E-0
  1.08554E-01  7.20601E-02

         AIRFOIL SPANWISE COORDINATE =  1.25000E+00  SECTION LIFT COEFFICIENT =  0.15534E+00 -1.43157E+00

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 -6.25347E+00  1.08440E+00 -6.43649E+00  1.30608E+00 -8.07895E+00  1.21336E+00 -3.00601E+00  1.10340E+00 -3.64572E+00  1.10287E+0
 -3.37003E+00  1.03047E+00 -2.64301E+00  7.03043E-01 -2.86713E+00  8.74006E-01 -3.46504E+00  1.33510E+00 -5.54581E+00  1.08253E+0
 -1.00250E+01  1.24598E+00 -1.30038E+01 -2.38593E-01  6.01977E+00 -4.24806E-01  2.34528E+00 -2.37487E-01  7.79776E-01 -8.70814E-0
  2.10179E-01 -2.03137E-02

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  6.25347E+00 -1.08698E+00  4.03449E+00 -1.30688E+00  4.07895E+00 -1.21336E+00  3.00601E+00 -1.14390E+00  3.64572E+00 -1.10287E+0
  3.37003E+00 -1.03097E+00  2.64381E+00 -7.03091E-01  2.86713E+00 -8.74006E-01  3.94504E+00 -1.33519E+00  5.54581E+00 -1.08253E+0
  1.00259E+01 -1.24988E+00  1.30038E+01  2.38593E-01 -6.01977E+00  6.24906E-01 -2.39528E+00  2.37487E-01 -7.79776E-01  8.70819E-0
 -2.10179E-01  2.03137E-02

         AIRFOIL SPANWISE COORDINATE =  1.50000E+00  SECTION LIFT COEFFICIENT =  7.13910E+00 -1.60794E+00
```

● FINAL OUTPUT FOR RUN 6U (CONT'D)

```
AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 -5.93413E+00  1.67540E+00 -4.18325E+00  1.15009E+00 -3.82015E+00  1.00100E+00 -3.53376E+00  9.91604E-01 -3.33433E+00  9.42585E-0
 -3.05275E+00  8.71598E-01 -2.50137E+00  6.92943E-01 -1.10450E+00  8.70887E-01 -4.36899E+00  1.27801E+00 -5.61407E+00  1.56747E+0
 -6.26792E+00  2.92267E-01 -3.57804E+00 -7.74927E-01 -1.78961E-01 -3.19601E-01 -4.02240E-02 -1.80819E-01  2.91949E-02 -9.43059E-0
  8.12433E-02 -3.66355E-02

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  5.93413E+00 -1.67500E+00  4.18325E+00 -1.15009E+00  3.82015E+00 -1.00100E+00  3.53376E+00 -9.91604E-01  3.33433E+00 -9.42585E-0
  3.05275E+00 -8.71598E-01  2.50137E+00 -6.92943E-01  3.10450E+00 -8.70887E-01  9.36899E+00 -1.27801E+00  5.61407E+00 -1.56747E+0
  6.26792E+00 -2.92267E-01  3.57804E+00  7.74927E-01  1.78961E-01  3.19601E-01  9.02240E-02  1.80819E-01 -2.91949E-02  9.43059E-0
 -4.12933E-02  3.66355E-02

         AIRFOIL SPANWISE COORDINATE =  1.75000E+00  SECTION LIFT COEFFICIENT =  5.58643E+00 -1.10639E+00

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 -5.27780E+00  1.35611E+00 -3.66037E+00  9.14941E-01 -3.27310E+00  8.20421E-01 -2.95779E+00  7.50930E-01 -2.71078E+00  7.04114E-0
 -2.69799E+00  6.52429E-01 -2.24830E+00  5.75598E-01 -2.58090E+00  6.38300E-01 -5.05895E+00  9.60840E-01 -3.87772E+00  4.06516E-0
 -7.67574E-01 -3.25197E-01 -7.77080E-01 -2.03729E-01 -3.81598E-01 -1.39286E-01 -1.78533E-01 -1.03438E-01  1.64750E-02 -7.72066E-0
  7.15221E-02 -3.06027E-02

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  5.27780E+00 -1.35611E+00  3.66037E+00 -9.14941E-01  3.27310E+00 -8.20421E-01  2.95779E+00 -7.50930E-01  2.71078E+00 -7.04114E-0
  2.69799E+00 -6.52428E-01  2.24830E+00 -5.75598E-01  2.58090E+00 -6.38300E-01  5.05895E+00 -9.60840E-01  3.87772E+00 -4.06516E-0
  7.67574E-01  3.25197E-01  7.77080E-01  2.03729E-01  3.81598E-01  1.39286E-01  1.78533E-01  1.03438E-01 -1.64750E-02  7.72066E-0
 -7.15221E-02  3.06027E-02

         AIRFOIL SPANWISE COORDINATE =  1.90000E+00  SECTION LIFT COEFFICIENT =  9.15309E+00 -8.49770E-01

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 -4.37440E+00  1.04709E+00 -2.94162E+00  6.87709E-01 -2.53730E+00  6.05800E-01 -2.22332E+00  5.42427E-01 -1.99492E+00  4.95356E-0
 -1.84442E+00  4.60166E-01 -1.78795E+00  4.32648E-01 -2.04331E+00  4.16857E-01 -2.34737E+00  3.53799E-01 -1.79125E+00  1.17520E-0
 -9.01649E-01 -7.20185E-02 -5.82679E-01 -8.56536E-02 -3.20704E-01 -8.02052E-02 -1.28710E-01 -7.47801E-02  9.15015E-02 -7.36898E-0
  1.30181E-01 -4.39057E-02

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  4.37440E+00 -1.04709E+00  2.94162E+00 -6.87704E-01  2.53730E+00 -6.05800E-01  2.22332E+00 -5.42427E-01  1.99492E+00 -4.95356E-0
  1.84442E+00 -4.60166E-01  1.78795E+00 -4.32648E-01  2.04333E+00 -4.16857E-01  2.34737E+00 -3.53799E-01  1.79125E+00 -1.17520E-0
  9.01649E-01  7.20105E-02  5.82679E-01  8.56536E-02  3.20704E-01  8.02052E-02  1.28710E-01  7.47801E-02 -9.15015E-02  7.36848E-0
 -1.30101E-01  4.39057E-02

         AIRFOIL SPANWISE COORDINATE =  2.00000E+00  SECTION LIFT COEFFICIENT =  2.78197E+00 -5.56909E-01

AIRFOIL PRESSURE COEFFICIENTS, UPPER =
 -3.16244E+00  7.23673E-01 -2.02660E+00  4.56319E-01 -1.60691E+00  3.91742E-01 -1.45161E+00  3.46470E-01 -1.29594E+00  3.16916E-0
 -1.20519E+00  2.91753E-01 -1.17884E+00  2.70140E-01 -1.22949E+00  2.27228E-01 -1.21542E+00  1.65250E-01 -9.70777E-01  6.68829E-0
 -6.90004E-01 -7.90813E-03 -8.21787E-01 -3.41582E-02 -2.36348E-01 -4.42040E-02 -8.87566E-02 -4.90640E-02  1.04062E-01 -5.09915E-0
  1.34871E-01 -3.93054E-02

AIRFOIL PRESSURE COEFFICIENTS, LOWER =
  3.16244E+00 -7.23673E-01  2.02660E+00 -4.56319E-01  1.60691E+00 -3.91742E-01  1.45161E+00 -3.46470E-01  1.29594E+00 -3.16916E-0
  1.20519E+00 -2.91753E-01  1.17884E+00 -2.70140E-01  1.22949E+00 -2.27228E-01  1.21542E+00 -1.65250E-01  9.70777E-01 -6.68829E-0
  6.90004E-01  7.90813E-03  8.21787E-01  3.41582E-02  2.36348E-01  4.42040E-02  8.87566E-02  4.90640E-02 -1.04062E-01  5.09915E-0
 -1.34871E-01  3.93054E-02
```

```
*** 3-D CIRCULAR ARC ***

MACH NUMBER = 8.00000E-01
SIMILARITY PARAMETER = 1.15900E+00
THICKNESS RATIO = 6.00000E-02
AIRFOIL ANGLE OF ATTACK (RADIANS) = 0.
REDUCED FREQUENCY (BASED ON CHORD) = 1.00000E-01
SCALED FREQUENCY (OMEGA) = 3.29173E-01
PITCH AXIS (XP) = 0.
WING ASPECT RATIO = 2.00000E+00
CP SCALING FACTOR (CP/CPBAR) = 1.26500E-01

UNSTEADY FORCE COEFFICIENTS (PER UNIT PITCH ANGLE IN RADIANS)

  LIFT = 1.27049E+01 -5.08702E+00
  MOMENT ABOUT (X=XP) = 5.18061E+00 -9.67302E-01

          AIRFOIL SPANWISE COORDINATE = 0.          SECTION LIFT COEFFICIENT = 9.19540E+00 -2.50649E+00

PRESSURE COEFFICIENTS ON THE AIRFOIL (PER UNIT PITCH ANGLE IN RADIANS)

 AIRFOIL STREAMWISE COORDINATE
  0.                 7.00000E-02        1.40000E-01        2.10000E-01        2.80000E-01
  3.50000E-01        4.20000E-01        5.00000E-01        5.50000E-01        6.00000E-01
  6.50000E-01        7.00000E-01        7.60000E-01        8.20000E-01        9.00000E-01
  1.00000E+00

 AIRFOIL PRESSURE COEFFICIENTS, UPPER =
  -6.63692E+00  2.25370E+00 -4.73147E+00  1.57457E+00 -4.37673E+00  1.47132E+00 -4.11974E+00  1.39450E+00 -3.99762E+00  1.36229E+0
  -3.73633E+00  1.20214E+00 -2.84761E+00  9.37414E-01 -2.85523E+00  8.00020E-01 -3.55076E+00  1.13977E+00 -2.69540E+00  1.05751E+0
  -5.19040E+00  1.94204E+00 -9.79159E+00  1.90619E+00 -6.27650E+00  3.09413E-01 -1.33114E+00 -9.72575E-02 -8.06382E-01 -7.06077E-0
  -2.82616E-01 -3.82769E-02

 AIRFOIL PRESSURE COEFFICIENTS, LOWER =
   6.63692E+00 -2.25370E+00  4.73147E+00 -1.57457E+00  4.37673E+00 -1.47132E+00  4.11974E+00 -1.39450E+00  3.99762E+00 -1.36229E+0
   3.73633E+00 -1.20214E+00  2.84761E+00 -9.37414E-01  2.85523E+00 -8.00020E-01  3.55076E+00 -1.13977E+00  2.69540E+00 -1.05751E+0
   5.19040E+00 -1.94204E+00  9.79159E+00 -1.90619E+00  6.27650E+00 -3.09413E-01  1.33114E+00  9.72575E-02  8.06382E-01  7.06077E-0
   2.82616E-01  3.82769E-02
```

8.0 REFERENCES

1. Traci, R.M., Albano, E.D., Farr, Jr., J.L., Cheng, H.K., "Small Disturbance Transonic Flows About Oscillating Airfoils", AFFDL-TR-74-30, April 1974.

2. Farr, Jr., J.L., Traci, R.M., Albano, E.D., "Computer Programs for Calculating Small Disturbance Transonic Flows About Oscillating Airfoils", AFFDL-TR-74-135, November 1974.

3. Traci, R.M., Albano, E.D., Farr, Jr., J.L., "Small Disturbance Transonic Flows About Oscillating Airfoils and Planar Wings", AFFDL-TR-75-100, August 1975.

4. Ehlers, F.E., "A Finite Difference Method for the Solutions of the Transonic Flow Around Harmonically Oscillating Wings," NASA CR-2257, January 1974.

5. Murman, E.M., and Cole, J.D., "Calculation of Plane Steady Transonic Flows," AIAA Paper 70-188, June 1970.

6. Krupp, J.A., "The Numerical Calculation of Plane Steady Transonic Flows Past Thin Lifting Airfoils," Boeing Scientific Research Laboratories, D180-12958-1, June 1971.

7. Klunker, E.B., "Contribution to Methods for Calculating the Flow about Thin Lifting Wings at Transonic Speeds," NASA TN D-6530, November 1971.

8. Bailey, F.R., and Steger, J.L., "Relaxation Techniques for Three-Dimensional Transonic Flows About Wings," AIAA Paper 72-189, San Diego, California, 1972.

9. Newman, P.A., and Klunker, E.B., "Computation of Transonic Flow About Finite Lifting Wings," AIAA Journal Vol. 10, No. 7, p. 971, 1972.

APPENDIX A

FORTRAN LISTING OF TDSTRN

A FORTRAN listing of the source deck for the TDSTRN program is presented in the following pages. The program, as configured here, requires 70.5_8K words to load and 57.0_8K words to execute. In this configuration the programs fit into small core of the CDC 7600.

```
      PROGRAM TDSTRN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,
     1  TAPE8)
      REAL KCAP,M8,IWING
      COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
     4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
      COMMON /INTERP/ ZE(25),NZE
      DIMENSION PHIDC(40),PHIDG(40),OMEGA(40),V(40),EPSGRD(3)
      NAMELIST /IN/ X,Y,Z,IM,JM,KM,ILE,ITE,JW,KSPAN,ZSPAN,M8,GAM,DEL,
     1  ALPHA,GAMFF,OMEGAH,OMEGAE,OMEGAP,EPSCOL,EPSGRD,NDUMP,NCOL,
     2  NGRID,NGFF,PGFF,KEPS,IK,NPRINT,NKPRT,KPRT,ZE,NZE
      NAMELIST /CONTRL/ ITAPE
      DATA AL,BET /.5,.5/
C
C  TO RESTART PROGRAM, COPY THE DATA FOR RESTART FROM TAPE7 TO A DISC
C  FILE TAPE8, POSITION TAPE7 AT THE END OF THE LAST FILE ON THE TAPE
C  SO NEW DATA MAY BE WRITTEN ON THE TAPE WITHOUT LOSING ANY OF THE
C  OLD DATA
C
      READ (5,912) (TITLE(I),I=1,8)
      READ (5,CONTRL)
      IF (ITAPE.EQ.0) GO TO 10
C  READ DATA FROM RESTART TAPE
      READ (8) NITERG,IM,IM1,JM,JM1,KM,KM1,JW,JWP1,JWM1,ITE,ILE,
     1  KSPAN,KCAP,DEL,ALPHA,NDB,M8,GAM,DYBU1,DYBU2,DYBL1,DYBL2,
     2  DOUB,ZSPAN
      READ (8) (X(I),DX(I),AX1(I),AX2(I),BX1(I),BX2(I),CX(I),I=1,IM)
      READ (8) (Y(I),DY(I),AY1(I),AY2(I),I=1,JM)
      READ (8) (Z(I),DZ(I),AZ1(I),AZ2(I),I=1,KM)
      L=ITE*KM
      READ (8) (FPU(I),FPL(I),PHIUB(I),I=1,L)
      READ (8) (GAMTE(I),GAMFF(I),I=1,KSPAN)
      L=IM*JM*KM
      READ (8) (PHI(I),I=1,L)
      IK=0
      READ (5,IN)
C  THE IK OPTION IS USED TO BOOT STRAP TO DIFFERENT MACH NUMBERS,
C  AIRFOIL THICKNESSES AND/OR ANGLES OF ATTACK MAKE SURE YOU HAVE INPUT
C  THE NEW M8, DEL AND/OR ALPHA
      IF (IK.EQ.0) GO TO 1
      KCAP=(1.-M8**2)/((1.+GAM)*DEL*M8**2)**.6666666667
      CALL INITAL
      CALL FARFLD
    1 CONTINUE
      SK=SQRT(KCAP)
      DO 2 I=1,KSPAN
      GAMTE1(I)=GAMTE(I)
    2 CONTINUE
      WRITE (6,900)
      WRITE (6,901) NITERG
      NITERG=0
```

```
      GO TO 15
C   START PROBLEM FROM SCRATCH
   10 CONTINUE
      READ (5,IN)
      KCAP=(1.-MB**2)/((1.+GAM)*DEL*MB**2)**.6666666667
      SK=SQRT(KCAP)
      DO 3 I=1,KSPAN
      GAMTE(I)=GAMFF(I)
      GAMTE1(I)=GAMFF(I)
    3 CONTINUE
      NITERG=0
      NDB=0
      IM1=IM-1
      JM1=JM-1
      KM1=KM-1
      JWP1=JW+1
      JWM1=JW-1
C   INITIALIZE FINITE DIFFERENCE COEFFICIENTS AND FARFIELD
      CALL INITAL
      CALL FARFLD
C   INITIAL GUESS FOR SUBSONIC CASE (INTERIOR ONLY)
      DO 20 K=1,KM1
      MP=IM*JM*(K-1)
      Z2=Z(K)**2
      DO 30 I=2,IM1
      M=MP+(I-1)*JM
      X2=X(I)**2
      CON=-X(I)*DOUB/(6.2831853)
      DO 40 J=2,JM1
      L=M+J
      R=SQRT(X2+KCAP*(Y(J)**2+Z2))
      IF (R.EQ.0.) GO TO 41
      PHI(L)=CON/R**3
      IF (ABS(PHI(L)).GT.1.) PHI(L)=SIGN(1.,X(I))
      GO TO 40
   41 CONTINUE
      PHI(L)=PHI(L-JM)
   40 CONTINUE
   30 CONTINUE
   20 CONTINUE
      L=ITE*KM
      DO 5 I=1,L
      PHIUB(I)=0.
    5 CONTINUE
      M=(ILE-2)*JM+JW
      KK=(ILE-1)*KM
      DO 47 K=1,KSPAN
      L=M+IM*JM*(K-1)
      PHIUB(KK+K)=PHI(L)
   47 CONTINUE
   15 CONTINUE
      WRITE (6,IN)
      WRITE (6,900)
      CPCPB=DEL**.6666666667/((1.+GAM)*MB**2)**.3333333333
      WRITE (6,913) KCAP,CPCPB
```

```
      KGRD=1
C  RE-CYCLE POINT FOR GRID ITERATION
   75 CONTINUE
      ERROR=0.
      NIT=NITERG
      NITERG=NITERG+1
      IF (MOD(NITERG,NPRINT).EQ.0) CALL PRINT(NIT)
      IF (MOD(NITERG,NGFF).NE.0) GO TO 76
      CALL GAMFUN
      CALL FARFLD
      WRITE (6,910) NITERG,IWING,GAMTE(1),GAMFF(1),GAMTE(KSPAN),
     1  GAMFF(KSPAN)
   76 CONTINUE
C  BEGIN LOOP ON THE PLANES (Z DIRECTION)
      IMJM=IM*JM
      DO 100 K=1,KM1
      MP=IMJM*(K-1)
C  CHECK FOR AIRFOIL (CORRECT PLANE)
      IFOIL=0
      IF (K.LE.KSPAN) IFOIL=1
C  BEGIN LOOP ON A GIVEN PLANE (X DIRECTION)
      DO 200 I=2,IM1
      M=MP+(I-1)*JM
C  CHECK FOR AIRFOIL (CORRECT COLUMN)
      IFLAG=0
      IF (IFOIL.EQ.1.AND.ILE.LE.I.AND.I.LE.ITE) IFLAG=1
      IF (IFLAG.EQ.1) N=(I-1)*KM+K
C  SAVE THIS COLUMN OF PHI
      DO 201 J=2,JM1
      L=M+J
      PHIOG(J)=PHI(L)
  201 CONTINUE
      NITERC=0
C  LOOP BACK POINT FOR COLUMN ITERATION
  250 CONTINUE
      NITERC=NITERC+1
      IF (NITERC.GT.NCOL) GO TO 394
C  SAVE PREVIOUS PHI FOR COLUMN ITERATION
      DO 202 J=2,JM1
      L=M+J
      PHIOC(J)=PHI(L)
  202 CONTINUE
C  BEGIN LOOP ON COLUMN (Y DIRECTION)
      DO 300 J=2,JM1
C  CALCULATE CELL INDICES
      L=M+J
      LR=L+JM
      LL=L-JM
      LLL=LL-JM
      IF (I.EQ.2) LLL=LL
      LA=L+1
      LB=L-1
      LF=L+IMJM
      LBK=L-IMJM
      IF (K.EQ.1) LBK=LF
```

```
      TPHIR=PHI(LR)
      TPHIL=PHI(LL)
      TPHILL=PHI(LLL)
      TPHIBK=PHI(LBK)
      IF (IFOIL.EQ.0.OR.J.NE.JW) GO TO 301
      IF (I.EQ.ILE-1) PHI(LR)=.5*(PHIUB((ILE-1)*KM+K)+PHI(LR))
      IF (I.EQ.ITE+1) PHI(LL)=.5*(PHIUB((ITE-1)*KM+K)+PHI(LL))
      IF (I.EQ.ITE+1) PHI(LLL)=.5*(PHIUB((ITE-2)*KM+K)+PHI(LLL))
      IF (I.EQ.ITE+2) PHI(LLL)=.5*(PHIUB((ITE-1)*KM+K)+PHI(LLL))
  301 CONTINUE
      IF (ILE.LE.I.AND.I.LE.ITE.AND.J.EQ.JW.AND.K.EQ.KSPAN+1)
     1  PHI(LBK)=.5*(PHIUB((I-1)*KM+KSPAN)+PHI(LBK))
      V(J)=KCAP-AX1(I-1)*(PHI(L)-PHI(LL))-AX2(I-1)*(PHI(LL)-PHI(LLL))
C  SET UP TRIDIAGONAL MATRIX TO SOLVE FOR PHI(I,J,K)
C  A * PHI(I,J+1,K) + B * PHI(I,J,K) + C * PHI(I,J-1,K) = D
      IF (IFLAG.EQ.1.AND.J.EQ.JWP1) GO TO 350
      IF (IFLAG.EQ.1.AND.J.EQ.JW) GO TO 360
      IF (IFLAG.EQ.1.AND.J.EQ.JWM1) GO TO 370
      PART=0.
      IF (I.LE.ITE.OR.IFOIL.EQ.0) GO TO 302
C  KUTTA CONDITION
      SIGI=(X(I)-1.)*(GAMFF(K)-GAMTE(K))/(X(IM1)-1.)+GAMTE(K)
      IF (J.EQ.JWM1) PART=.5*AY1(J)*SIGI
      IF (J.EQ.JW) PART=.5*(AY1(J)-AY2(J))*SIGI
      IF (J.EQ.JWP1) PART=-.5*AY2(J)*SIGI
  302 CONTINUE
      VV=KCAP-AX1(I)*(PHI(LR)-PHI(L))-AX2(I)*(PHI(L)-PHI(LL))
      IF (VV.LT.0.) GO TO 320
      IF (V(J).LT.0.) GO TO 380
C  ****************************************************************************
C  *                          ELLIPTIC DIFFERENCING
C  ****************************************************************************
      OMEGA(J)=OMEGAF
      A(J)=AY1(J)
      B(J)=-(VV*(BX1(I)+BX2(I))+AY1(J)+AY2(J))-AZ1(K)-AZ2(K)
      C(J)=AY2(J)
      D(J)=-VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))+PART-(AZ1(K)*PHI(LF)+
     1  AZ2(K)*PHI(LBK))
      IF (J.EQ.2) GO TO 303
      IF (J.EQ.JM1) GO TO 304
      GO TO 390
C  BOTTOM BOUNDARY
  303 CONTINUE
      D(J)=D(J)-AY2(J)*PHI(LB)
      GO TO 390
C  TOP BOUNDARY
  304 CONTINUE
      D(J)=D(J)-AY1(J)*PHI(LA)
      GO TO 390
  320 CONTINUE
      IF (V(J).GT.0.) GO TO 340
C  ****************************************************************************
C  *                        HYPERBOLIC DIFFERENCING
C  ****************************************************************************
      OMEGA(J)=OMEGAH
```

```
      VV=KCAP-CX(I-1)*(PHI(L)-PHI(LL))-CX(I-2)*(PHI(LL)-PHI(LLL))
      A(J)=AY1(J)
      B(J)=VV*BX1(I-1)-AY1(J)-AY2(J)-AZ1(K)-AZ2(K)
      C(J)=AY2(J)
      D(J)=VV*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-PHI(LLL)))+PART-
     1  (AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      IF (J.EQ.2) GO TO 322
      IF (J.EQ.JM1) GO TO 323
      GO TO 390
C  BOTTOM BOUNDARY
  322 CONTINUE
      D(J)=D(J)-AY2(J)*PHI(LB)
      GO TO 390
C  TOP BOUNDARY
  323 CONTINUE
      D(J)=D(J)-AY1(J)*PHI(LA)
      GO TO 390
C  *********************************************************************
C  *                        PARABOLIC DIFFERENCING
C  *********************************************************************
  340 CONTINUE
      OMEGA(J)=OMEGAP
      A(J)=AY1(J)
      B(J)=VV*BX1(I-1)-AY1(J)-AY2(J)-AZ1(K)-AZ2(K)
      C(J)=AY2(J)
      D(J)=VV*(BX1(I-1)+BX2(I-1))*PHI(LL)-VV*BX2(I-1)*PHI(LLL)+PART-
     1  (AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      IF (J.EQ.2) GO TO 342
      IF (J.EQ.JM1) GO TO 343
      GO TO 390
C  BOTTOM BOUNDARY
  342 CONTINUE
      D(J)=D(J)-AY2(J)*PHI(LB)
      GO TO 390
C  TOP BOUNDAY
  343 CONTINUE
      D(J)=D(J)-AY1(J)*PHI(LA)
      GO TO 390
C  *********************************************************************
C  *                      SHOCK POINT DIFFERENCING
C  *********************************************************************
  380 CONTINUE
      OMEGA(J)=OMEGAE
      A(J)=AY1(J)
      B(J)=-AL*VV*(BX1(I)+BX2(I))+BET*V(J)*BX1(I-1)-AY1(J)-AY2(J)-
     1  AZ1(K)-AZ2(K)
      C(J)=AY2(J)
      D(J)=-AL*VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))+BET*V(J)*(BX1(I-1)*
     1  PHI(LL)+BX2(I-1)*(PHI(LL)-PHI(LLL)))-(AZ1(K)*PHI(LF)+AZ2(K)*
     2  PHI(LBK))
      GO TO 390
C  *********************************************************************
C  *               AIRFOIL UPPER SURFACE BOUNDARY CONDITION
C  *********************************************************************
  350 CONTINUE
```

```
      VV=KCAP-AX1(I)*(PHI(LR)-PHI(L))-AX2(I)*(PHI(L)-PHI(LL))
      IF (VV.LT.0.) GO TO 351
      IF (V(J).LT.0.) GO TO 353
C  ELLIPTIC
      OMEGA(J)=OMEGAE
      A(J)=DYBU1
      B(J)=-(DYBU1+VV*(BX1(I)+BX2(I)))-AZ1(K)-AZ2(K)
      C(J)=0.
      D(J)=DYBU2*FPU(N)-VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))-(AZ1(K)*
     1  PHI(LF)+AZ2(K)*PHI(LBK))
      GO TO 390
  351 CONTINUE
      IF (V(J).GT.0.) GO TO 352
C  HYPERBOLIC
      OMEGA(J)=OMEGAH
      VV=KCAP-CX(I-1)*(PHI(L)-PHI(LL))-CX(I-2)*(PHI(LL)-PHI(LLL))
      A(J)=DYBU1
      B(J)=VV*BX1(I-1)-DYBU1-AZ1(K)-AZ2(K)
      C(J)=0.
      D(J)=DYBU2*FPU(N)+VV*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-
     1  PHI(LLL)))-(AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      GO TO 390
C  PARABOLIC
  352 CONTINUE
      OMEGA(J)=OMEGAP
      A(J)=DYBU1
      B(J)=VV*BX1(I-1)-DYBU1-AZ1(K)-AZ2(K)
      C(J)=0.
      D(J)=DYBU2*FPU(N)+VV*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-
     1  PHI(LLL)))-(AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      GO TO 390
C  SHOCK POINT
  353 CONTINUE
      OMEGA(J)=OMEGAE
      A(J)=DYBU1
      B(J)=-DYBU1-AL*VV*(BX1(I)+BX2(I))+BET*V(J)*BX1(I-1)-AZ1(K)-AZ2(K)
      C(J)=0.
      D(J)=DYBU2*FPU(N)-AL*VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))+BET*V(J)*
     1  (BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-PHI(LLL)))-(AZ1(K)*PHI(LF)+
     2  AZ2(K)*PHI(LBK))
      GO TO 390
C  ***************************************************************************
C  *                AIRFOIL LOWER SURFACE BOUNDARY CONDITION                  *
C  ***************************************************************************
  370 CONTINUE
      VV=KCAP-AX1(I)*(PHI(LR)-PHI(L))-AX2(I)*(PHI(L)-PHI(LL))
      IF (VV.LT.0.) GO TO 371
      IF (V(J).LT.0.) GO TO 373
C  ELLIPTIC
      OMEGA(J)=OMEGAE
      A(J)=0.
      B(J)=-(DYBL1+VV*(BX1(I)+BX2(I)))-AZ1(K)-AZ2(K)
      C(J)=DYBL1
      D(J)=-DYBL2*FPL(N)-VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))-(AZ1(K)*
     1  PHI(LF)+AZ2(K)*PHI(LBK))
```

```
      GO TO 390
  371 CONTINUE
      IF (V(J).GT.0.) GO TO 372
C  HYPERBOLIC
      OMEGA(J)=OMEGAH
      VV=KCAP-CX(I-1)*(PHI(L)-PHI(LL))-CX(I-2)*(PHI(LL)-PHI(LLL))
      A(J)=0.
      B(J)=VV*BX1(I-1)-DYBL1-AZ1(K)-AZ2(K)
      C(J)=DYBL1
      D(J)=-DYBL2*FPL(N)+VV*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-
     1  PHI(LLL)))-(AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      GO TO 390
C  PARABOLIC
  372 CONTINUE
      OMEGA(J)=OMEGAP
      A(J)=0.
      B(J)=VV*BX1(I-1)-DYBL1-AZ1(K)-AZ2(K)
      C(J)=DYBL1
      D(J)=-DYBL2*FPL(N)+VV*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-
     1  PHI(LLL)))-(AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      GO TO 390
C  SHOCK POINT
  373 CONTINUE
      OMEGA(J)=OMEGAE
      A(J)=0.
      B(J)=-DYBL1-AL*VV*(BX1(I)+BX2(I))+BET*V(J)*BX1(I-1)-AZ1(K)-AZ2(K)
      C(J)=DYBL1
      D(J)=-DYBL2*FPL(N)-AL*VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))+BET*
     1  V(J)*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-PHI(LLL)))-(AZ1(K)*
     2  PHI(LF)+AZ2(K)*PHI(LBK))
      GO TO 390
  360 CONTINUE
C  BODY BOUNDARY J=JW
      A(J)=0.
      B(J)=1.
      C(J)=0.
      D(J)=PHI(L)
  390 CONTINUE
      PHI(LR)=TPHIR
      PHI(LL)=TPHIL
      PHI(LLL)=TPHILL
      PHI(LBK)=TPHIBK
  300 CONTINUE
C  TRIDIAGONAL MATRIX IS SET NOW SOLVE FOR COLUMN OF PHI
      CALL TRI(I,K)
C  CHECK FOR COLUMN CONVERGENCE OF PHI
      DO 395 J=2,JM1
      L=M+J
      JERROR=J
      ERRC=PHIOC(J)-PHI(L)
      IF (ABS(ERRC).GT.EPSCOL) GO TO 250
  395 CONTINUE
  394 CONTINUE
      IF (NITERC.GT.NCOL) WRITE (6,904) NITERG,I,ERRC,JERROR,K
C  CONVERGED, RELAX PHI, FIND ERROR AND MOVE TO NEXT COLUMN
```

```
      DO 396 J=2,JM1
      L=M+J
      ERR=OMEGA(J)*(PHI(L)-PHIOG(J))
      PHI(L)=PHIOG(J)+ERR
      IF (ABS(ERR).LT.ABS(ERROR)) GO TO 396
      ERROR=ERR
      LERROR=L
  396 CONTINUE
      IF (IFLAG.NE.1) GO TO 200
      L=M+JW
      PHI(L)=PHI(L-1)+DY(JWM1)*(PHI(L-1)-PHI(L-2))/DY(JW-2)
      PHIUB(N)=PHI(L+1)-DY(JW)*(PHI(L+2)-PHI(L+1))/DY(JWP1)
      IF (I.EQ.ITE) GAMTE(K)=PHIUB(N)-PHI(L)
  200 CONTINUE
  100 CONTINUE
C  PRINT OUT ERROR AFTER GRID SWEEP
      WRITE (6,905) NITERG,ERROR,LERROR
      IF (ABS(ERROR).LT.10.) GO TO 101
      WRITE (6,915)
      STOP
  101 CONTINUE
      IDOUB=0
      IF (ABS(ERROR).LE.EPSGRD(KGRD)) GO TO 400
      IF (NITERG.EQ.NGRID) GO TO 410
      IF (MOD(NITERG,NDUMP).EQ.0) GO TO 410
      GO TO 75
  400 CONTINUE
      KGRD=KGRD+1
      IDOUB=1
      GO TO 410
  401 CONTINUE
      CALL GAMFUN
      WRITE (6,910) NITERG,IWING,GAMTE(1),GAMFF(1),GAMTE(KSPAN),
     1  GAMFF(KSPAN)
      CALL FPRINT
      WRITE (6,900)
      WRITE (6,906) NITERG
      CALL DOUBLE
      WRITE (6,914) IM,JM,JW,KM,ILE,ITE,KSPAN
      WRITE (6,902)
      WRITE (6,903) (X(I),I=1,IM)
      WRITE (6,911)
      WRITE (6,903) (Y(I),I=1,JM)
      WRITE (6,916)
      WRITE (6,903) (Z(I),I=1,KM)
      GO TO 75
  410 CONTINUE
C  TAPE DUMP
      WRITE (7) NITERG,IM,IM1,JM,JM1,KM,KM1,JW,JWP1,JWM1,ITE,ILE,
     1  KSPAN,KCAP,DEL,ALPHA,NDB,MB,GAM,DYBU1,DYBU2,DYBL1,DYBL2,
     2  DOUB,ZSPAN
      WRITE (7) (X(I),DX(I),AX1(I),AX2(I),BX1(I),BX2(I),CX(I),I=1,IM)
      WRITE (7) (Y(I),DY(I),AY1(I),AY2(I),I=1,JM)
      WRITE (7) (Z(I),DZ(I),AZ1(I),AZ2(I),I=1,KM)
      L=ITE*KM
```

```
      WRITE (7) (FPU(I),FPL(I),PHIUB(I),I=1,L)
      WRITE (7) (GAMTE(I),GAMFF(I),I=1,KSPAN)
      L=IM*JM*KM
      WRITE (7) (PHI(I),I=1,L)
      END FILE 7
      WRITE (6,907) NITERG
      CALL PRINT(NITERG)
      IF (KGRD.GT.KEPS) GO TO 420
      IF (NITERG.EQ.NGRID) GO TO 430
      IF (IDOUB.EQ.1) GO TO 401
      GO TO 75
  420 CONTINUE
      WRITE (6,908) NITERG
      GO TO 450
  430 CONTINUE
      WRITE (6,909) NITERG
  900 FORMAT (1H1)
  901 FORMAT (1H ,/,* CASE IS BEING RESTARTED AT ITERATION*I5)
  902 FORMAT (1H ,/,* X(I),I=1,IM*)
  903 FORMAT (10E13.5)
  904 FORMAT (1H ,/,* AT ITERATION*I5* COLUMN*I4* FAILED TO CONVERGE*
     1  * ERR =*E13.5* J =*I3* K =*I3)
  905 FORMAT (1H ,/,* AT ITERATION*I5* THE MAXIMUM ERROR =*E13.5* AND OC
     1CURRED AT NODE*I5)
  906 FORMAT (1H ,/,* THE NUMBER OF NODES IS BEING DOUBLED AT ITERATION*
     1  I5)
  907 FORMAT (1H ,/,* TAPE HAS BEEN DUMPED AT ITERATION*I5)
  908 FORMAT (1H ,/,* SOLUTION HAS CONVERGED TO DESIRED ACCURACY AT ITER
     1ATION*I5)
  909 FORMAT (1H ,/,* MAXIMUM NUMBER OF ITERATIONS HAS BEEN REACHED, CAS
     1E IS BEING TERMINATED AT ITERATION*I5)
  910 FORMAT (1H ,/,* UPDATE GAMFF AND FARFIELD AT ITERATION*I5,/,
     1  * IWING =*E13.5* GAMTE(1) =*E13.5* GAMFF(1) =*E13.5* GAMTE(KSPAN
     2) =*E13.5* GAMFF(KSPAN) =*E13.5)
  911 FORMAT (1H ,/,* Y(J),J=1,JM*)
  912 FORMAT (8A10)
  913 FORMAT (1H ,/,* SIMILARITY PARAMETER (K) =*E13.5,/,* SCALING FACTO
     1R (CP/CPBAR) =*E13.5)
  914 FORMAT (1H ,/,* IM =*I4* JM =*I4* JW =*I4* KM =*I4* ILE =*I4
     1  * ITE =*I4* KSPAN =*I4)
  915 FORMAT (1H ,/,* SOLUTION IS DIVERGING, THE PROBLEM IS BEING TERMIN
     1ATED*)
  916 FORMAT (1H ,/,* Z(K),K=1,KM*)
  450 CONTINUE
      CALL FPRINT
      END
      SUBROUTINE DOUBLE
      REAL KCAP,M8,IWING
      COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
     4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
```

```
      RETURN
      END
      SUBROUTINE FARFLD
      REAL KCAP,M8,IWING
      COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
     4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
      COMMON /INTERP/ ZE(25),NZE
C   SUBSONIC FARFIELD
C   CALCULATE WING INTEGRAL
      CON1=DOUB/6.2831853
      IWING=0.
      DO 10 I=2,KSPAN
      IWING=IWING+.5*(GAMTE(I)+GAMTE(I-1))*DZ(I-1)
   10 CONTINUE
      IWING=2.*(IWING+.5*GAMTE(KSPAN)*(ZSPAN-Z(KSPAN)))
      CON2=IWING/12.5663706
C   Z=Z(KM)
      MP=IM*JM*KM1
      Z2=Z(KM)**2
      DO 20 I=1,IM1
      M=MP+(I-1)*JM
      X2=X(I)**2
      CON3=-X(I)*CON1
      DO 25 J=1,JM
      L=M+J
      Y2=Y(J)**2
      R=SQRT(X2+KCAP*(Y2+Z2))
      PHIT=CON3/R**3
      PHI(L)=PHIT+CON2*Y(J)*(1.+X(I)/R)/(Y2+Z2)
   25 CONTINUE
   20 CONTINUE
C   X=X(1)
      X2=X(1)**2
      CON3=-X(1)*CON1
      DO 30 K=1,KM1
      MP=IM*JM*(K-1)
      Z2=Z(K)**2
      DO 35 J=1,JM
      L=MP+J
      Y2=Y(J)**2
      R=SQRT(X2+KCAP*(Y2+Z2))
      PHIT=CON3/R**3
      IF (Y2.EQ.0..AND.Z2.EQ.0.) GO TO 36
      PHI(L)=PHIT+CON2*Y(J)*(1.+X(1)/R)/(Y2+Z2)
      GO TO 35
   36 CONTINUE
      PHI(L)=PHIT
   35 CONTINUE
   30 CONTINUE
C   Y=Y(1) AND Y=Y(JM)
```

```
      DO 40 ID=1,2
      IF (ID.EQ.2) GO TO 41
      J=1
      Y2=Y(1)**2
      GO TO 42
   41 CONTINUE
      J=JM
      Y2=Y(JM)**2
   42 CONTINUE
      DO 45 K=1,KM1
      MP=IM*JM*(K-1)
      Z2=Z(K)**2
      CON3=CON2*Y(J)/(Y2+Z2)
      DO 46 I=2,IM1
      L=MP+(I-1)*JM+J
      R=SQRT(X(I)**2+KCAP*(Y2+Z2))
      PHIT=-X(I)*CON1/R**3
      PHI(L)=PHIT+CON3*(1.+X(I)/R)
   46 CONTINUE
   45 CONTINUE
   40 CONTINUE
C     X=X(IM)
      X2=X(IM)**2
      DO 50 K=1,KM
      MP=IM*JM*(K-1)
      M=MP+IM1*JM
      Z2=Z(K)**2
      DO 70 I=1,NZE
      A(I)=ZE(I)
   70 CONTINUE
      NEND=NZE
      IFLIP=0
      IZ=0
      IF (Z(K).LE.ZSPAN) IZ=1
      DO 60 J=1,JM
      IF (ABS(Y(J)).LE..5.OR.IFLIP.EQ.1) GO TO 71
      DO 72 I=1,KSPAN
      A(I)=Z(I)
   72 CONTINUE
      NEND=KSPAN
      IFLIP=1
   71 CONTINUE
      L=M+J
      Y2=Y(J)**2
      CON3=Y(J)/3.14159265
      AINT=0.
      IF (IZ.EQ.1.AND.J.EQ.JW) GO TO 65
      IZE=2
      DO 61 KK=1,NEND
   62 CONTINUE
      IF (A(KK).LE.Z(IZE)) GO TO 63
      IZE=IZE+1
      GO TO 62
   63 CONTINUE
      ANEW=(GAMTE(IZE-1)+(A(KK)-Z(IZE-1))/DZ(IZE-1)*(GAMTE(IZE)-
```

```
  1  GAMTE(IZE-1)))/((Z(K)-A(KK))**2+Y2)
     IF (KK.EQ.1) GO TO 64
     AINT=AINT+.5*(ANEW+OLD)*(A(KK)-A(KK-1))
 64  CONTINUE
     OLD=ANEW
 61  CONTINUE
     AINT=AINT+.5*ANEW*(ZSPAN-Z(KSPAN))
 65  CONTINUE
     R=SQRT(X2+KCAP*(Y2+Z2))
     PHIT=-X(IM)*CON1/R**3
     PHI(L)=PHIT+CON3*AINT
 60  CONTINUE
 50  CONTINUE
     RETURN
     END
     FUNCTION FLP(XX,ZZ)
     REAL KCAP,M8,IWING
     COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
    1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
    2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
    3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
    4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
C  AIRFOIL LOWER SURFACE SLOPE DISTRIBUTION
     FLP=4.*(XX-.5)
     RETURN
     END
     SUBROUTINE FPRINT
     REAL KCAP,M8,IWING
     COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
    1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
    2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
    3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
    4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
     COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
     COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
     CPCPB=DEL**.6666666667/((1.+GAM)*M8**2)**.3333333333
     CPCRIT=-2.*CPCPB*KCAP
     WRITE (6,900)
     WRITE (6,901) (TITLE(I),I=1,8)
     WRITE (6,902) M8
     WRITE (6,903) KCAP
     WRITE (6,904) DEL
     WRITE (6,905) ALPHA
     WRITE (6,907) ZSPAN
     WRITE (6,906) CPCPB
     WRITE (6,916) CPCRIT
     WRITE (6,912)
     WRITE (6,915) (X(I),I=ILE,ITE)
     CLIFT=0.
     CMOM=0.
     DO 10 K=1,KSPAN
     PART=.5*((X(ITE+1)-1.)*(GAMFF(K)-GAMTE(K))/(X(IM1)-1.)+GAMTE(K))
     MP=IM*JM*(K-1)
     IJK=MP+ITE*JM+JW
     PHI(IJK)=PHI(IJK)-PART
```

```
      L=MP+(ILE-2)*JM+JW
      LP=(ILE-2)*KM+K
      PHIUB(LP)=PHI(L)
      LP=ITE*KM+K
      PHIUB(LP)=PHI(IJK)+2.*PART
      DO 20 I=ILE,ITE
      M=MP+(I-1)*JM
      L=M+JW
      LP=(I-1)*KM+K
      A(I)=-2.*(AX1(I)*(PHI(L+JM)-PHI(L))+AX2(I)*(PHI(L)-PHI(L-JM)))*
     1  CPCPB
      B(I)=-2.*(AX1(I)*(PHIUB(LP+KM)-PHIUB(LP))+AX2(I)*(PHIUB(LP)-
     1  PHIUB(LP-KM)))*CPCPB
      IF (K.GT.1) GO TO 21
      C(I)=A(I)
      D(I)=B(I)
   21 CONTINUE
      C1=A(I)-B(I)
      C2=C1*X(I)
      IF (I.GT.ILE) GO TO 22
      CL=C1*X(ILE)
      CM=.5*C2*X(ILE)
      GO TO 23
   22 CONTINUE
      CL=CL+.5*(C1+C10)*DX(I-1)
      CM=CM+.5*(C2+C20)*DX(I-1)
   23 CONTINUE
      C10=C1
      C20=C2
   20 CONTINUE
      PHI(IJK)=PHI(IJK)+PART
      IF (K.EQ.1) GO TO 11
      CLIFT=CLIFT+.5*(CL+CLO)*DZ(K-1)
      CMOM=CMOM+.5*(CM+CMO)*DZ(K-1)
   11 CONTINUE
      CLO=CL
      CMO=CM
      DO 12 N=1,NKPRT
      IF (KPRT(N).NE.K) GO TO 12
      GAMPRT=2.*GAMTE(K)*CPCPB
      WRITE (6,908) Z(K),GAMPRT
      WRITE (6,913)
      WRITE (6,915) (B(I),I=ILE,ITE)
      WRITE (6,914)
      WRITE (6,915) (A(I),I=ILE,ITE)
      GO TO 10
   12 CONTINUE
   10 CONTINUE
      WRITE (6,900)
      WRITE (6,901) (TITLE(I),I=1,8)
      WRITE (6,902) MB
      WRITE (6,903) KCAP
      WRITE (6,904) DEL
      WRITE (6,905) ALPHA
      WRITE (6,907) ZSPAN
```

```
      WRITE (6,906) CPCPB
      WRITE (6,916) CPCRIT
      WRITE (6,909)
      WRITE (6,910) CLIFT,CMOM
      GAMPRT=2.*GAMTE(1)*CPCPB
      WRITE (6,908) Z(1),GAMPRT
      WRITE (6,912)
      WRITE (6,915) (X(I),I=ILE,ITE)
      WRITE (6,913)
      WRITE (6,915) (D(I),I=ILE,ITE)
      WRITE (6,914)
      WRITE (6,915) (C(I),I=ILE,ITE)
  900 FORMAT (1H1)
  901 FORMAT (30X,8A10)
  902 FORMAT (1H ,/,1H ,/,1H ,/,* MACH NUMBER =*E13.5)
  903 FORMAT (* SIMILARITY PARAMETER (K) =*E13.5)
  904 FORMAT (* THICKNESS RATIO =*E13.5)
  905 FORMAT (* AIRFOIL ANGLE OF ATTACK (RADIANS) =*E13.5)
  906 FORMAT (* CP SCALING FACTOR (CP/CPBAR) =*E13.5)
  907 FORMAT (* WING ASPECT RATIO =*E13.5)
  908 FORMAT (1H ,/,1H ,/,21X*AIRFOIL SPANWISE COORDINATE =*E13.5
     1  *  SECTION LIFT COEFFICIENT =*E13.5)
  909 FORMAT (1H ,/,1H ,/,* AIRFOIL FORCE COEFFICIENTS*)
  910 FORMAT (1H ,/,3X*LIFT =*E13.5,/,3X*MOMENT ABOUT (X=0) =*E13.5)
  912 FORMAT (1H ,/,1H ,/,3X*AIRFOIL STREAMWISE COORDINATE*)
  913 FORMAT (1H ,/,1H ,/,3X*AIRFOIL PRESSURE COEFFICIENTS, UPPER =*)
  914 FORMAT (1H ,/,3X*AIRFOIL PRESSURE COEFFICIENTS, LOWER =*)
  915 FORMAT (3X10E13.5)
  916 FORMAT (* CRITICAL PRESSURE COEFFICIENT (SONIC) =*E13.5)
      RETURN
      END
      FUNCTION FUP(XX,ZZ)
      REAL KCAP,M8,IWING
      COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
     4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
C  AIRFOIL UPPER SURFACE SLOPE DISTRIBUTION
      FUP=-4.*(XX-.5)
C  DOUB IS THE DOUBLET STRENGTH DUE TO THICKNESS
      DOUB=1.333333333*ZSPAN
      RETURN
      END
      SUBROUTINE GAMFUN
      REAL KCAP,M8,IWING
      COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
     4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
      DO 10 I=1,KSPAN
      GAMFF(I)=GAMTE1(I)+PGFF*(GAMTE(I)-GAMTE1(I))
      GAMTE1(I)=GAMTE(I)
```

```
   10 CONTINUE
      RETURN
      END
      SUBROUTINE INITAL
      REAL KCAP,M8,IWING
      COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
     4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
C  CALCULATE DX,DY AND DZ
      DO 10 I=1,IM1
      DX(I)=X(I+1)-X(I)
   10 CONTINUE
      DO 20 I=1,JM1
      DY(I)=Y(I+1)-Y(I)
   20 CONTINUE
      DO 30 I=1,KM1
      DZ(I)=Z(I+1)-Z(I)
   30 CONTINUE
C  FINITE DIFFERENCE COEFFICIENTS
      DO 40 I=2,IM1
      AX1(I)=DX(I-1)/(DX(I)*(DX(I-1)+DX(I)))
      AX2(I)=DX(I)/(DX(I-1)*(DX(I-1)+DX(I)))
      BX1(I)=2.*AX1(I)/DX(I-1)
      BX2(I)=2.*AX2(I)/DX(I)
      CX(I)=.5/DX(I)
   40 CONTINUE
      CX(1)=.5/DX(1)
      DO 50 I=2,JM1
      AY1(I)=2./(DY(I)*(DY(I)+DY(I-1)))
      AY2(I)=2./(DY(I-1)*(DY(I)+DY(I-1)))
   50 CONTINUE
      DO 60 I=2,KM1
      AZ1(I)=2./(DZ(I)*(DZ(I)+DZ(I-1)))
      AZ2(I)=2./(DZ(I-1)*(DZ(I)+DZ(I-1)))
   60 CONTINUE
      AZ1(1)=2./DZ(1)**2
      AZ2(1)=0.
      DYBU1=2./((DY(JWP1)+2.*DY(JW))*DY(JWP1))
      DYBU2=DY(JWP1)*DYBU1
      DYBL1=2./((DY(JW-2)+2.*DY(JWM1))*DY(JW-2))
      DYBL2=DY(JW-2)*DYBL1
C  SET AIRFOIL BOUNDARY CONDITION
      DO 70 K=1,KSPAN
      DO 80 I=ILE,ITE
      L=(I-1)*KM+K
      FPU(L)=FUP(X(I),Z(K))
      FPL(L)=FLP(X(I),Z(K))
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE PRINT (NITERG)
      REAL KCAP,M8,IWING
```

```
      COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
     4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
      KSPAN1=KSPAN-1
      DO 10 K=1,KSPAN,KSPAN1
      PART=.5*((X(ITE+1)-1.)*(GAMFF(K)-GAMTE(K))/(X(IM1)-1.)+GAMTE(K))
      MP=IM*JM*(K-1)
      IJK=MP+ITE*JM+JW
      PHI(IJK)=PHI(IJK)-PART
      L=MP+(ILE-2)*JM+JW
      LP=(ILE-2)*KM+K
      PHIUB(LP)=PHI(L)
      LP=ITE*KM+K
      PHIUB(LP)=PHI(IJK)+2.*PART
C   COMPUTE CP LOWER (A) AND CP UPPER (B)
      DO 20 I=ILE,ITE
      L=MP+(I-1)*JM+JW
      LP=(I-1)*KM+K
      A(I)=-2.*(AX1(I)*(PHI(L+JM)-PHI(L))+AX2(I)*(PHI(L)-PHI(L-JM)))
      B(I)=-2.*(AX1(I)*(PHIUB(LP+KM)-PHIUB(LP))+AX2(I)*(PHIUB(LP)-
     1  PHIUB(LP-KM)))
   20 CONTINUE
      PHI(IJK)=PHI(IJK)+PART
      WRITE (6,901) NITERG,K
      WRITE (6,902) (B(I),I=ILE,ITE)
      WRITE (6,903) NITERG,K
      WRITE (6,902) (A(I),I=ILE,ITE)
   10 CONTINUE
  901 FORMAT (1H ,/,* AT ITERATION*I5* AND K =*I3* SCALED PRESSURE COEFF
     1ICIENT, UPPER (ILE TO ITE) =*)
  902 FORMAT (10E13.5)
  903 FORMAT (1H ,/,* AT ITERATION*I5* AND K =*I3* SCALED PRESSURE COEFF
     1ICIENT, LOWER (ILE TO ITE) =*)
      RETURN
      END
      SUBROUTINE TRI (I,K)
      REAL KCAP,M8,IWING
      COMMON /DELTA/ DX(40),DY(40),DZ(20),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,ALPHA,DEL,M8,
     4  GAM,KCAP,NDB,TITLE(8),DOUB,IWING,ZSPAN,NKPRT,KPRT(20)
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      MP=IM*JM*(K-1)
      DO 10 KK=3,JM1
      J=JM1-KK+3
      P=A(J-1)/B(J)
      B(J-1)=B(J-1)-P*C(J)
      D(J-1)=D(J-1)-P*D(J)
   10 CONTINUE
      M=MP+(I-1)*JM
```

```
      PHI(M+2)=D(2)/B(2)
      DO 20 J=3,JM1
      L=M+J
      PHI(L)=(D(J)-PHI(L-1)*C(J))/B(J)
   20 CONTINUE
      RETURN
      END
```

APPENDIX B

FORTRAN LISTING OF TDUTRN

A FORTRAN listing of the source deck for the TDUTRN program is presented in the following pages. The program, as configured here, requires 161.7_8K words to load and 150.0_8K words to execute. In this configuration the programs fit into small core of the CDC 7600.

```
      PROGRAM TDUTRN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,
     1  TAPE8,TAPE9)
      COMPLEX B,D,PHIUB,PHI,PHIOG,GAMTE1,GAMTE,GAMFF,ERR,ERROR,
     1  OMEG2I,SIGI,PART,FPU,FPL,TPHIR,TPHIL,TPHILL,TPHIBK,CON
      REAL KCAP,M8
      COMMON /DELTA/ DX(40),DY(40),DZ(40),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,SMALLK,OMEG,
     4  NDOUB,CPCP8,TITLE(8),M8,DEL,ALPHA,ITYPE,IOPT,XP,NKPRT,KPRT(20),
     5  ZSPAN,KCAP,RPAR
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
      COMMON /STEADY/ PHIS(11500),AX1S(40),AX2S(40),BX1S(40),BX2S(40),
     1  CXS(40),PHIUBS(800)
      COMMON /INTERP/ ZE(25),NZE
      DIMENSION PHIOG(40),OMEGA(40),V(40),EPSGRD(3)
      NAMELIST /IN/ X,Y,Z,IM,JM,KM,ILE,ITE,JW,KSPAN,GAMFF,OMEGAH,
     1  OMEGAE,OMEGAP,EPSGRD,NDUMP,NGRID,NGFF,PGFF,KEPS,NPRINT,NKPRT,
     2  KPRT,SMALLK,IK,XP,ITYPE,IOPT,ZE,NZE
      NAMELIST /CONTRL/ ITAPE
C
C   TO START PROGRAM, STEADY (PHIS) DATA IS TO BE READ FROM A DISC FILE
C   TAPE8.  UNSTEADY DATA WILL NOW BE WRITTEN ON TAPE7.
C
C   TO RESTART PROGRAM, AGAIN STORE STEADY DATA AS ABOVE AND STORE
C   THE UNSTEADY DATA ON A DISC FILE TAPE9.  NEW UNSTEADY DATA WILL
C   NOW BE WRITTEN ON TAPE7.
C
C  READ STEADY SOLUTION
      READ (8) DUM,IMS,IM1S,JMS,JM1S,KMS,KM1S,JWS,DUM,DUM,ITES,ILES,
     1  KSPANS,KCAP,DEL,ALPHA,NDB,M8,GAM,DUM,DUM,DUM,DUM,DUM,ZSPAN
      READ (8) (DUM,DUM,AX1S(I),AX2S(I),BX1S(I),BX2S(I),CXS(I),I=1,IMS)
      READ (8) DUM
      READ (8) DUM
      L=ITES*KMS
      READ (8) (DUM,DUM,PHIUBS(I),I=1,L)
      READ (8) DUM
      L=IMS*JMS*KMS
      READ (8) (PHIS(I),I=1,L)
C  MODIFY LEADING AND TRAILING EDGE PHI
      M=IMS*JMS
      MC1=(ILES-1)*JMS+JWS
      MC2=(ITES-1)*JMS+JWS
      MC3=(ITES-2)*JMS+JWS
      LP1=(ILES-1)*KMS
      LP2=(ITES-1)*KMS
      LP3=(ITES-2)*KMS
      DO 1 K=1,KSPANS
      MP=M*(K-1)
      L=MP+MC1
      LP=LP1+K
      PHIS(L)=.5*(PHIS(L)+PHIUBS(LP))
      L=MP+MC2
      LP=LP2+K
```

```
      PHIS(L)=.5*(PHIS(L)+PHIUBS(LP))
      L=MP+MC3
      LP=LP3+K
      PHIS(L)=.5*(PHIS(L)+PHIUBS(LP))
    1 CONTINUE
      SK=SQRT(KCAP)
      CPCPB=DEL**.6666666667/((1.+GAM)*M8**2)**.3333333333
      RPAR=1./((1.+GAM)*M8**2*DEL)**.3333333333
      READ (5,911) (TITLE(I),I=1,8)
      READ (5,CONTRL)
      IF (ITAPE.EQ.0) GO TO 10
C   READ DATA FROM RESTART TAPE
      READ (9) NITERG,IM,IM1,JM,JM1,KM,KM1,JW,JWP1,JWM1,ITE,ILE,
     1  KSPAN,OMEG,SMALLK,DYBU1,DYBU2,DYBL1,DYBL2,NDOUB,XP
      READ (9) (X(I),DX(I),AX1(I),AX2(I),BX1(I),BX2(I),CX(I),I=1,IM)
      READ (9) (Y(I),DY(I),AY1(I),AY2(I),I=1,JM)
      READ (9) (Z(I),DZ(I),AZ1(I),AZ2(I),I=1,KM)
      L=ITE*KM
      READ (9) (FPU(I),FPL(I),PHIUB(I),I=1,L)
      READ (9) (GAMTE(I),GAMFF(I),I=1,KSPAN)
      L=IM*JM*KM
      READ (9) (PHI(I),I=1,L)
      DO 2 I=1,KSPAN
      GAMTE1(I)=GAMTE(I)
    2 CONTINUE
      IK=0
      READ (5,IN)
      WRITE (6,900)
      WRITE (6,901) NITERG
      NITERG=0
      WRITE (6,913) KCAP,CPCPB
C   THE IK OPTION IS USED TO BOOTSTRAP TO DIFFERENT REDUCED FREQUENCIES
C   AND/OR MODES OF OSCILLATION
      IF (IK.EQ.0) GO TO 15
      OMEG=SMALLK*M8**2/((1.+GAM)*DEL*M8**2)**.6666666667
      CALL INITAL
      CALL FARFLD
      GO TO 15
C   START PROBLEM FROM SCRATCH
   10 CONTINUE
      READ (5,IN)
      NITERG=0
      NDOUB=0
      OMEG=SMALLK*M8**2/((1.+GAM)*DEL*M8**2)**.6666666667
      DO 3 I=1,KSPAN
      GAMTE1(I)=GAMFF(I)
      GAMTE(I)=GAMFF(I)
    3 CONTINUE
      IM1=IM-1
      JM1=JM-1
      KM1=KM-1
      JWP1=JW+1
      JWM1=JW-1
C   INITIALIZE FINITE DIFFERENCE COEFFICIENTS AND FARFIELD
      CALL INITAL
```

```
      CALL FARFLD
C  INITIALIZE GUESS FOR SUBSONIC CASE (INTERIOR ONLY)
C  ASSUMED SYMMETRY IN Y
      JW2=2*JW
      CON=GAMFF(1)/6.2831853
      DO 20 K=1,KM1
      MP=IM*JM*(K-1)
      ZP=Z(K)+ZSPAN
      ZM=Z(K)-ZSPAN
      Z2=Z(K)**2
      DO 30 I=2,IM1
      M=MP+(I-1)*JM
      PHI(M+JW)=CMPLX(0.,0.)
      X2=X(I)**2
      DO 40 J=JWP1,JM1
      L=M+J
      LL=M+JW2-J
      Y2=Y(J)**2
      IF (X(I).LT.1.) GO TO 41
      PHI(L)=CON*(ATAN(ZP/Y(J))-ATAN(ZM/Y(J)))
      GO TO 42
   41 CONTINUE
      R=SQRT(X2+KCAP*(Y2+Z2))
      PHI(L)=CON*ZSPAN*(Y(J)/(Y2+Z2))*(1.+X(I)/R)
   42 CONTINUE
      CPHI=CABS(PHI(L))
      IF (CPHI.GT.1.) PHI(L)=PHI(L)/CPHI
      PHI(LL)=-PHI(L)
   40 CONTINUE
   30 CONTINUE
   20 CONTINUE
      L=ITE*KM
      ERR=CMPLX(0.,0.)
      DO 4 I=1,L
      PHIUB(I)=ERR
    4 CONTINUE
      M=(ILE-2)*JM+JW
      KK=(ILE-1)*KM
      DO 45 K=1,KSPAN
      L=M+IM*JM*(K-1)
      PHIUB(KK+K)=PHI(L)
   45 CONTINUE
   15 CONTINUE
      OMEG2I=CMPLX(0.,2.*OMEG)
      WRITE (6,IN)
      WRITE (6,900)
      IF (ITAPE.EQ.0) WRITE (6,913) KCAP,CPCPB
      KGRD=1
C  RE-CYCLE POINT FOR GRID ITERATION
   50 CONTINUE
      ERROR=CMPLX(0.,0.)
      NIT=NITERG
      NITERG=NITERG+1
      IF (MOD(NITERG,NPRINT).EQ.0) CALL PRINT(NIT)
      IF (MOD(NITERG,NGFF).NE.0) GO TO 51
```

```
      CALL GAMFUN
      CALL FARFLD
      WRITE (6,910) NITERG,GAMTE(1),GAMFF(1),GAMTE(KSPAN),GAMFF(KSPAN)
   51 CONTINUE
      INCR=2**(NDB-NDOUB)
      KS=1-INCR
      IMJM=IM*JM
      IMSJMS=IMS*JMS
C   BEGIN LOOP ON THE PLANES (Z DIRECTION)
      DO 100 K=1,KM1
      KS=KS+INCR
      DO 102 I=2,JM1
      V(I)=KCAP
  102 CONTINUE
      MP=IMJM*(K-1)
      MPS=IMSJMS*(KS-1)
C   CHECK FOR AIRFOIL
      IFOIL=0
      IF (K.LE.KSPAN) IFOIL=1
      IS=2-INCR
C   BEGIN LOOP ON A GIVEN PLANE (X DIRECTION)
      DO 200 I=2,IM1
      IS=IS+INCR
C   CHECK FOR AIRFOIL
      IFLAG=0
      IF (IFOIL.EQ.1.AND.ILE.LE.I.AND.I.LE.ITE) IFLAG=1
      IF (IFLAG.EQ.1) N=(I-1)*KM+K
      M=MP+(I-1)*JM
      MS=MPS+(IS-1)*JMS
C   SAVE THIS COLUMN OF PHI
      DO 201 J=2,JM1
      L=M+J
      PHIOG(J)=PHI(L)
  201 CONTINUE
      JS=2-INCR
C   BEGIN LOOP ON COLUMN (Y DIRECTION)
      DO 300 J=2,JM1
      JS=JS+INCR
C   CALCULATE CELL INDICES FOR PHIO
      LS=MS+JS
      LSR=LS+JMS
      LSL=LS-JMS
      LSLL=LSL-JMS
      IF (IS.EQ.2) LSLL=LSL
C   CALCULATE CELL INDICES FOR PHI1
      L=M+J
      LR=L+JM
      LL=L-JM
      LLL=LL-JM
      IF (I.EQ.2) LLL=LL
      LB=L-1
      LA=L+1
      LF=L+IMJM
      LBK=L-IMJM
      IF (K.EQ.1) LBK=LF
```

```
C  ************************************************************************
C  *                 CALCULATE V AND PHIXX FROM STEADY SOLUTION
C  ************************************************************************
      VV=KCAP-AX1S(IS)*(PHIS(LSR)-PHIS(LS))-AX2S(IS)*(PHIS(LS)-
     1  PHIS(LSL))
      VVS=VV
      IF (VV.LT.0.) GO TO 301
C  ELLIPTIC
      OMEGA(J)=OMEGAE
      PHIXX=BX1S(IS)*(PHIS(LSR)-PHIS(LS))-BX2S(IS)*(PHIS(LS)-PHIS(LSL))
      GO TO 302
  301 CONTINUE
      OMEGA(J)=OMEGAP
      IF (V(J).GT.0.) GO TO 303
C  HYPERBOLIC
      OMEGA(J)=OMEGAH
      VV=KCAP-CXS(IS-1)*(PHIS(LS)-PHIS(LSL))-CXS(IS-2)*(PHIS(LSL)-
     1  PHIS(LSLL))
  303 CONTINUE
C  PARABOLIC
      PHIXX=BX1S(IS-1)*(PHIS(LS)-PHIS(LSL))-BX2S(IS-1)*(PHIS(LSL)-
     1  PHIS(LSLL))
  302 CONTINUE
      V(J)=VVS
      TPHIR=PHI(LR)
      TPHIL=PHI(LL)
      TPHILL=PHI(LLL)
      TPHIBK=PHI(LBK)
      IF (IFOIL.EQ.0.OR.J.NE.JW) GO TO 304
      IF (I.EQ.ILE-1) PHI(LR)=.5*(PHIUB((ILE-1)*KM+K)+PHI(LR))
      IF (I.EQ.ITE+1) PHI(LL)=.5*(PHIUB((ITE-1)*KM+K)+PHI(LL))
      IF (I.EQ.ITE+1) PHI(LLL)=.5*(PHIUB((ITE-2)*KM+K)+PHI(LLL))
      IF (I.EQ.ITE+2) PHI(LLL)=.5*(PHIUB((ITE-1)*KM+K)+PHI(LLL))
  304 CONTINUE
      IF (ILE.LE.I.AND.I.LE.ITE.AND.J.EQ.JW.AND.K.EQ.KSPAN+1)
     1  PHI(LBK)=.5*(PHIUB((I-1)*KM+KSPAN)+PHI(LBK))
C  SET UP TRIDIAGONAL MATRIX TO SOLVE FOR PHI(I,J,K)
C  A * PHI(I,J+1,K) + B * PHI(I,J,K) + C * PHI(I,J-1,K) = D
      IF (IFLAG.EQ.1.AND.J.EQ.JWP1) GO TO 330
      IF (IFLAG.EQ.1.AND.J.EQ.JW) GO TO 340
      IF (IFLAG.EQ.1.AND.J.EQ.JWM1) GO TO 350
      PART=CMPLX(0.,0.)
      IF (I.LE.ITE.OR.IFOIL.EQ.0) GO TO 305
C  KUTTA CONDITION
      SIGI=(X(I)-1.)*(GAMFF(K)-GAMTE(K))/(X(IM1)-1.)+GAMTE(K)
      IF (IOPT.EQ.1) SIGI=SIGI*CEXP(CMPLX(0.,-SMALLK*(X(I)-1.)))
      IF (J.EQ.JWM1) PART=.5*AY1(J)*SIGI
      IF (J.EQ.JW) PART=.5*(AY1(J)-AY2(J))*SIGI
      IF (J.EQ.JWP1) PART=-.5*AY2(J)*SIGI
  305 CONTINUE
      IF (VVS.LT.0.) GO TO 320
C  ************************************************************************
C  *                          ELLIPTIC DIFFERENCING
C  ************************************************************************
      A(J)=AY1(J)
```

```
      B(J)=-(VV*(BX1(I)+BX2(I))+AY1(J)+AY2(J)+(OMEG2I+PHIXX)*(AX2(I)-
     1  AX1(I)))-AZ1(K)-AZ2(K)
      C(J)=AY2(J)
      D(J)=-VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))+(OMEG2I+PHIXX)*(AX1(I)*
     1  PHI(LR)-AX2(I)*PHI(LL))+PART-(AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      IF (J.EQ.2) GO TO 311
      IF (J.EQ.JM1) GO TO 312
      GO TO 390
C  BOTTOM BOUNDARY
  311 CONTINUE
      D(J)=D(J)-AY2(J)*PHI(LB)
      GO TO 390
C  TOP BOUNDARY
  312 CONTINUE
      D(J)=D(J)-AY1(J)*PHI(LA)
      GO TO 390
C  ***************************************************************************
C  *                    HYPERBOLIC AND PARABOLIC DIFFERENCING
C  ***************************************************************************
  320 CONTINUE
      A(J)=AY1(J)
      B(J)=VV*BX1(I-1)-AY1(J)-AY2(J)-(OMEG2I+PHIXX)*CX(I-1)-AZ1(K)-
     1  AZ2(K)
      C(J)=AY2(J)
      D(J)=VV*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-PHI(LLL)))-(OMEG2I+
     1  PHIXX)*(CX(I-1)*PHI(LL)-CX(I-2)*(PHI(LL)-PHI(LLL)))+PART-
     2  (AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      IF (J.EQ.2) GO TO 321
      IF (J.EQ.JM1) GO TO 322
      GO TO 390
C  BOTTOM BOUNDARY
  321 CONTINUE
      D(J)=D(J)-AY2(J)*PHI(LB)
      GO TO 390
C  TOP BOUNDARY
  322 CONTINUE
      D(J)=D(J)-AY1(J)*PHI(LA)
      GO TO 390
C  ***************************************************************************
C  *                 AIRFOIL UPPER SURFACE BOUNDARY CONDITION
C  ***************************************************************************
  330 CONTINUE
      IF (VVS.LT.0.) GO TO 331
C  ELLIPTIC
      A(J)=DYBU1
      B(J)=-(VV*(BX1(I)+BX2(I))+DYBU1+(OMEG2I+PHIXX)*(AX2(I)-AX1(I)))-
     1  AZ1(K)-AZ2(K)
      C(J)=0.
      D(J)=DYBU2*FPU(N)-VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))+(OMEG2I+
     1  PHIXX)*(AX1(I)*PHI(LR)-AX2(I)*PHI(LL))-(AZ1(K)*PHI(LF)+
     2  AZ2(K)*PHI(LBK))
      GO TO 390
C  HYPERBOLIC AND PARABOLIC
  331 CONTINUE
      A(J)=DYBU1
```

```
      B(J)=VV*BX1(I-1)-DYBU1-(OMEG2I+PHIXX)*CX(I-1)-AZ1(K)-AZ2(K)
      C(J)=0.
      D(J)=DYBU2*FPU(N)+VV*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-
     1  PHI(LLL)))-(OMEG2I+PHIXX)*(CX(I-1)*PHI(LL)-CX(I-2)*(PHI(LL)-
     2  PHI(LLL)))-(AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      GO TO 390
C  ********************************************************************
C  *                AIRFOIL LOWER SURFACE BOUNDARY CONDITION
C  ********************************************************************
  350 CONTINUE
      IF (VVS.LT.0.) GO TO 351
C  ELLIPTIC
      A(J)=0.
      B(J)=-(DYBL1+VV*(BX1(I)+BX2(I))+(OMEG2I+PHIXX)*(AX2(I)-AX1(I)))-
     1  AZ1(K)-AZ2(K)
      C(J)=DYBL1
      D(J)=-DYBL2*FPL(N)-VV*(BX1(I)*PHI(LR)+BX2(I)*PHI(LL))+(OMEG2I+
     1  PHIXX)*(AX1(I)*PHI(LR)-AX2(I)*PHI(LL))-(AZ1(K)*PHI(LF)+
     2  AZ2(K)*PHI(LBK))
      GO TO 390
C  HYPERBOLIC AND PARABOLIC
  351 CONTINUE
      A(J)=0.
      B(J)=VV*BX1(I-1)-DYBL1-(OMEG2I+PHIXX)*CX(I-1)-AZ1(K)-AZ2(K)
      C(J)=DYBL1
      D(J)=-DYBL2*FPL(N)+VV*(BX1(I-1)*PHI(LL)+BX2(I-1)*(PHI(LL)-
     1  PHI(LLL)))-(OMEG2I+PHIXX)*(CX(I-1)*PHI(LL)-CX(I-2)*(PHI(LL)-
     2  PHI(LLL)))-(AZ1(K)*PHI(LF)+AZ2(K)*PHI(LBK))
      GO TO 390
C  BODY BOUNDARY J=JW
  340 CONTINUE
      A(J)=0.
      B(J)=CMPLX(1.,0.)
      C(J)=0.
      D(J)=PHI(L)
  390 CONTINUE
      PHI(LR)=TPHIR
      PHI(LL)=TPHIL
      PHI(LLL)=TPHILL
      PHI(LBK)=TPHIBK
      IF (IOPT.EQ.0) GO TO 300
      IF (IFLAG.EQ.1.AND.J.EQ.JW) GO TO 300
      B(J)=B(J)+SMALLK*OMEG
  300 CONTINUE
C  TRIDIAGONAL MATRIX IS SET NOW SOLVE FOR COLUMN OF PHI
      CALL TRI(I,K)
C  RELAX PHI, FIND ERROR AND MOVE TO NEXT COLUMN
      DO 395 J=2,JM1
      L=M+J
      ERR=OMEGA(J)*(PHI(L)-PHIOG(J))
      PHI(L)=PHIOG(J)+ERR
      IF (CABS(ERR).LT.CABS(ERROR)) GO TO 395
      ERROR=ERR
      LERROR=L
  395 CONTINUE
```

```
      IF (IFLAG.NE.1) GO TO 200
      L=M+JW
      PHI(L)=PHI(L-1)+DY(JWM1)*(PHI(L-1)-PHI(L-2))/DY(JW-2)
      PHIUB(N)=PHI(L+1)-DY(JW)*(PHI(L+2)-PHI(L+1))/DY(JWP1)
      IF (I.EQ.ITE) GAMTE(K)=PHIUB(N)-PHI(L)
  200 CONTINUE
  100 CONTINUE
C  PRINT OUT ERROR AFTER EACH GRID SWEEP
      WRITE (6,905) NITERG,ERROR,LERROR
      IF (CABS(ERR).LT.10.) GO TO 101
      WRITE (6,912)
      STOP
  101 CONTINUE
      IDOUB=0
      IF (CABS(ERROR).LE.EPSGRD(KGRD)) GO TO 400
      IF (NITERG.EQ.NGRID) GO TO 410
      IF (MOD(NITERG,NDUMP).EQ.0) GO TO 410
      GO TO 50
  400 CONTINUE
      KGRD=KGRD+1
      IDOUB=1
      GO TO 410
  401 CONTINUE
      CALL GAMFUN
      WRITE (6,910) NITERG,GAMTE(1),GAMFF(1),GAMTE(KSPAN),GAMFF(KSPAN)
      CALL FPRINT
      WRITE (6,900)
      WRITE (6,906) NITERG
      CALL DOUBLE
      WRITE (6,914) IM,JM,JW,KM,ILE,ITE,KSPAN
      WRITE (6,902)
      WRITE (6,903) (X(I),I=1,IM)
      WRITE (6,904)
      WRITE (6,903) (Y(I),I=1,JM)
      WRITE (6,915)
      WRITE (6,903) (Z(I),I=1,KM)
      GO TO 50
  410 CONTINUE
      WRITE (7) NITERG,IM,IM1,JM,JM1,KM,KM1,JW,JWP1,JWM1,ITE,ILE,
     1  KSPAN,OMEG,SMALLK,DYBU1,DYBU2,DYBL1,DYBL2,NDOUB,XP
      WRITE (7) (X(I),DX(I),AX1(I),AX2(I),BX1(I),BX2(I),CX(I),I=1,IM)
      WRITE (7) (Y(I),DY(I),AY1(I),AY2(I),I=1,JM)
      WRITE (7) (Z(I),DZ(I),AZ1(I),AZ2(I),I=1,KM)
      L=ITE*KM
      WRITE (7) (FPU(I),FPL(I),PHIUB(I),I=1,L)
      WRITE (7) (GAMTE(I),GAMFF(I),I=1,KSPAN)
      L=IM*JM*KM
      WRITE (7) (PHI(I),I=1,L)
      END FILE 7
      WRITE (6,907) NITERG
      CALL PRINT(NITERG)
      IF (KGRD.GT.KEPS) GO TO 420
      IF (NITERG.EQ.NGRID) GO TO 430
      IF (IDOUB.EQ.1) GO TO 401
      GO TO 50
```

```
420 CONTINUE
    WRITE (6,908) NITERG
    GO TO 450
430 CONTINUE
    WRITE (6,909) NITERG
900 FORMAT (1H1)
901 FORMAT (1H ,/,* CASE IS BEING RESTARTED AT ITERATION*I5)
902 FORMAT (1H ,/,* X(I),I=1,IM*)
903 FORMAT (10E13.5)
904 FORMAT (1H ,/,* Y(I),I=1,JM*)
905 FORMAT (1H ,/,* AT ITERATION*I5* THE MAXIMUM ERROR =*2E13.5* AND O
   1CCURRED AT NODE*I5)
906 FORMAT (1H ,/,* THE NUMBER OF NODES IS BEING DOUBLED AT ITERATION*
   1  I5)
907 FORMAT (1H ,/,* TAPE HAS BEEN DUMPED AT ITERATION*I5)
908 FORMAT (1H ,/,* SOLUTION HAS CONVERGED TO DESIRED ACCURACY AT ITER
   1ATION*I5)
909 FORMAT (1H ,/,* MAXIMUM NUMBER OF ITERATIONS HAS BEEN REACHED, CAS
   1E IS BEING TERMINATED AT ITERATION*I5)
910 FORMAT (1H ,/,* UPDATE GAMFF AND FARFIELD AT ITERATION*I5,4X* GAMT
   1E(1) =*2E13.5,4X* GAMFF(1) =*2E13.5,/,44X* GAMTE(KSPAN) =*2E13.5*
   2GAMFF(KSPAN) =*2E13.5)
911 FORMAT (8A10)
912 FORMAT (1H ,/,* SOLUTION IS DIVERGING, THE PROBLEM IS BEING TERMIN
   1ATED*)
913 FORMAT (1H ,/,* SIMILARITY PARAMETER (K) =*E13.5,/,* SCALING FACTO
   1R (CP/CPBAR) =*E13,5)
914 FORMAT (1H ,/,* IM =*I4* JM =*I4* JW =*I4* KM =*I4* ILE =*I4
   1  * ITE =*I4* KSPAN =*I4)
915 FORMAT (1H ,/,* Z(I),I=1,KM*)
450 CONTINUE
    CALL FPRINT
    END
    SUBROUTINE DOUBLE
    COMPLEX B,D,PHIUB,PHI,GAMTE1,GAMTE,GAMFF,FPU,FPL
    REAL KCAP,M8
    COMMON /DELTA/ DX(40),DY(40),DZ(40),AX1(40),AX2(40),BX1(40),
   1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
   2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
   3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,SMALLK,OMEG,
   4  NDOUB,CPCPB,TITLE(8),M8,DEL,ALPHA,ITYPE,IOPT,XP,NKPRT,KPRT(20),
   5  ZSPAN,KCAP,RPAR
    COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
    COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
    RETURN
    END
    SUBROUTINE FARFLD
    COMPLEX B,D,PHI,PHIUB,FPU,FPL,GAMTE1,GAMTE,GAMFF,P1,P10,PART1,
   1  PART10,OMK,WING,AMUK,WAKEIN,G1,G2,GAMTEI,CON4,CONS
    REAL KCAP,M8
    COMMON /DELTA/ DX(40),DY(40),DZ(40),AX1(40),AX2(40),BX1(40),
   1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
   2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
   3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,SMALLK,OMEG,
   4  NDOUB,CPCPB,TITLE(8),M8,DEL,ALPHA,ITYPE,IOPT,XP,NKPRT,KPRT(20),
```

```
     5  ZSPAN,KCAP,RPAR
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
      COMMON /INTERP/ ZE(25),NZE
C  SUBSONIC FARFIELD (ASSUMED SYMMETRY IN Y)
      SK=SQRT(KCAP)
      CON1=1./6.2831853
      CON2=KCAP*CON1
      CON3=OMEG/KCAP
      AMU=SQRT(OMEG*(CON3+SMALLK))
      AMUK=CMPLX(0.,AMU/SK)
      BETA2=1.-MS**2
      OMK=CMPLX(0.,-CON3)
      JW2=2*JW
      IMJM=IM*JM
C  CALCULATE PART OF WING INTEGRAL
      DO 10 I=ILE,ITE
      ML=(I-1)*JM+JW
      MU=(I-1)*KM
      CON4=CEXP(OMK*X(I))
      PART1=CMPLX(0.,0.)
      DO 20 K=1,KSPAN
      L=IMJM*(K-1)+ML
      LP=MU+K
      P1=PHIUB(LP)-PHI(L)
      IF (K.EQ.1) GO TO 21
      PART1=PART1+.5*(P1+P10)*DZ(K-1)
   21 CONTINUE
      P10=P1
   20 CONTINUE
      PART1=CON4*(PART1+.5*P1*(ZSPAN-Z(KSPAN)))
      IF (I.EQ.ILE) GO TO 11
      WING=WING+.5*(PART1+PART10)*DX(I-1)
   11 CONTINUE
      IF (I.EQ.ILE) WING=.5*PART1*X(ILE)
      PART10=PART1
   10 CONTINUE
C  INTEGRATE GAMTE
      GAMTEI=CMPLX(0.,0.)
      DO 15 K=2,KSPAN
      GAMTEI=GAMTEI+.5*(GAMTE(K)+GAMTE(K-1))*DZ(K-1)
   15 CONTINUE
      GAMTEI=GAMTEI+.5*GAMTE(KSPAN)*(ZSPAN-Z(KSPAN))
C  Z=Z(KM)
      MP=IMJM*KM1
      Z2=Z(KM)**2
      DO 30 I=1,IM1
      M=MP+(I-1)*JM
      X2=X(I)**2
      X0=X(I)-1.
      X02=X0**2
      PHI(M+JW)=CMPLX(0.,0.)
      CON4=CON1*CEXP(CMPLX(0.,-SMALLK*X0))*GAMTEI
      CON5=CON2*WING*CEXP(CMPLX(0.,CON3*X(I)))
      DO 31 J=JWP1,JM
```

```
      L=M+J
      LL=M+JW2-J
      Y2=Y(J)**2
      R=KCAP*(Y2+Z2)
      BR=SQRT(X02+R)
      SR=SQRT(R)
      T1=(M8*BR-X0)/BETA2
      RH=SR*RPAR
      U=T1/RH
      CALL WAKF (U,SMALLK,RH,WAKEIN)
      G1=KCAP*M8*CEXP(CMPLX(0.,-SMALLK*T1))/(BR*(BR-M8*X0))
      G2=WAKEIN/R
      BR=SQRT(X2+KCAP*(Y2+Z2))
      PHI(L)=CON4*Y(J)*(G1+G2)+CON5*Y(J)*(1.+AMUK*BR)*CEXP(-AMUK*BR)/
     1  BR**3
      PHI(LL)=-PHI(L)
   31 CONTINUE
   30 CONTINUE
C  X=X(1)
      X2=X(1)**2
      X0=X(1)-1.
      X02=X0**2
      CON4=CON1*CEXP(CMPLX(0.,-SMALLK*X0))*GAMTEI
      CON5=CON2*WING*CEXP(CMPLX(0.,CON3*X(1)))
      DO 60 K=1,KM1
      M=IMJM*(K-1)
      Z2=Z(K)**2
      PHI(M+JW)=CMPLX(0.,0.)
      DO 61 J=JWP1,JM
      L=M+J
      LL=M+JW2-J
      Y2=Y(J)**2
      R=KCAP*(Y2+Z2.
      BR=SQRT(X02+R)
      SR=SQRT(R)
      T1=(M8*BR-X0)/BETA2
      RH=SR*RPAR
      U=T1/RH
      CALL WAKE (U,SMALLK,RH,WAKEIN)
      G1=KCAP*M8*CEXP(CMPLX(0.,-SMALLK*T1))/(BR*(BR-M8*X0))
      G2=WAKEIN/R
      BR=SQRT(X2+KCAP*(Y2+Z2))
      PHI(L)=CON4*Y(J)*(G1+G2)+CON5*Y(J)*(1.+AMUK*BR)*CEXP(-AMUK*BR)/
     1  BR**3
      PHI(LL)=-PHI(L)
   61 CONTINUE
   60 CONTINUE
C  X=X(IM)
      IJ=IM1*JM
      X2=X(IM)**2
      X0=X(IM)-1.
      X02=X0**2
      CON4=CON1*CEXP(CMPLX(0.,-SMALLK*X0))
      CON5=CON2*WING*CEXP(CMPLX(0.,CON3*X(IM)))
      DO 43 K=1,KM
```

```
      M=IMJM*(K-1)+IJ
      PHI(M+JW)=CMPLX(0.,0.)
      Z2=Z(K)**2
      DO 70 I=1,NZE
      A(I)=ZE(I)
   70 CONTINUE
      NEND=NZE
      IFLIP=0
      DO 44 J=JWP1,JM
      IF(Y(J).LE..5.OR.IFLIP.EQ.1) GO TO 71
      DO 72 I=1,KSPAN
      A(I)=Z(I)
   72 CONTINUE
      NEND=KSPAN
      IFLIP=1
   71 CONTINUE
      L=M+J
      LL=M+JW2-J
      Y2=Y(J)**2
      PART1=CMPLX(0.,0.)
      IZE=2
      DO 45 KK=1,NEND
   47 CONTINUE
      IF (A(KK).LE.Z(IZE)) GO TO 48
      IZE=IZE+1
      GO TO 47
   48 CONTINUE
      R=KCAP*(Y2+(Z(K)-A(KK))**2)
      BR=SQRT(X02+R)
      SR=SQRT(R)
      T1=(M8*BR-X0)/BETA2
      RH=SR*RPAR
      U=T1/RH
      CALL WAKE (U,SMALLK,RH,WAKEIN)
      G1=KCAP*M8*CEXP(CMPLX(0.,-SMALLK*T1))/(BR*(BR-M8*X0))
      G2=WAKEIN/R
      P1=(GAMTE(IZE-1)+(A(KK)-Z(IZE-1))/DZ(IZE-1)*(GAMTE(IZE)-
     1  GAMTE(IZE-1)))*(G1+G2)
      IF (KK.EQ.1) GO TO 46
      PART1=PART1+.5*(P1+P10)*(A(KK)-A(KK-1))
   46 CONTINUE
      P10=P1
   45 CONTINUE
      PART1=CON4*Y(J)*(PART1+.5*P1*(ZSPAN-Z(KSPAN)))
      BR=SQRT(X2+KCAP*(Y2+Z2))
      PHI(L)=PART1+CON5*Y(J)*(1.+AMUK*BR)*CEXP(-AMUK*BR)/BR**3
      PHI(LL)=-PHI(L)
   44 CONTINUE
   43 CONTINUE
C   Y=Y(1) AND Y=Y(JM)
      J=JM
      Y2=Y(J)**2
      DO 53 K=1,KM1
      MP=IMJM*(K-1)
      Z2=Z(K)**2
```

```
      DO 54 I=2,IM1
      M=MP+(I-1)*JM
      L=M+J
      LL=M+JW2-J
      X2=X(I)**2
      X0=X(I)-1.
      X02=X0**2
      CON4=CON1*CEXP(CMPLX(0.,-SMALLK*X0))*GAMTEI
      CON5=CON2*WING*CEXP(CMPLX(0.,CON3*X(I)))
      R=KCAP*(Y2+Z2)
      BR=SQRT(X02+R)
      SR=SQRT(R)
      T1=(MB*BR-X0)/BETA2
      RH=SR*RPAR
      U=T1/RH
      CALL WAKE (U,SMALLK,RH,WAKEIN)
      G1=KCAP*MB*CEXP(CMPLX(0.,-SMALLK*T1))/(BR*(BR-MB*X0))
      G2=WAKEIN/R
      BR=SQRT(X2+KCAP*(Y2+Z2))
      PHI(L)=CON4*Y(J)*(G1+G2)+CON5*Y(J)*(1.+AMUK*BR)*CEXP(-AMUK*BR)/
     1  BR**3
      PHI(LL)=-PHI(L)
   54 CONTINUE
   53 CONTINUE
      RETURN
      END
      SUBROUTINE FPRINT
      COMPLEX B,D,PHIUB,PHI,GAMTE1,GAMTE,GAMFF,FPU,FPL,PART,C1,C2,
     1  C10,C20,CL,CM,CLO,CMO,CLIFT,CMOM,GAMPRT,B1,D1
      REAL KCAP,MB
      COMMON /DELTA/ DX(40),DY(40),DZ(40),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,SMALLK,OMEG,
     4  NDOUB,CPCPB,TITLE(8),MB,DEL,ALPHA,ITYPE,IOPT,XP,NKPRT,KPRT(20),
     5  ZSPAN,KCAP,RPAR
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
      DIMENSION B1(40),D1(40)
      CPDEL=CPCPB/DEL
      WRITE (6,900)
      WRITE (6,901) (TITLE(I),I=1,8)
      WRITE (6,902) MB
      WRITE (6,903) KCAP
      WRITE (6,904) DEL
      WRITE (6,905) ALPHA
      WRITE (6,906) SMALLK
      WRITE (6,907) OMEG
      WRITE (6,908) XP
      WRITE (6,909) ZSPAN
      WRITE (6,910) CPCPB
      WRITE (6,911)
      WRITE (6,912) (X(I),I=ILE,ITE)
      CLIFT=CMPLX(0.,0.)
      CMOM=CMPLX(0.,0.)
```

```
      DO 10 K=1,KSPAN
      PART=.5*((X(ITE+1)-1.)*(GAMFF(K)-GAMTE(K))/(X(IM1)-1.)+GAMTE(K))
      IF (IOPT.EQ.1) PART=PART*CEXP(CMPLX(0.,-SMALLK*(X(ITE+1)-1.)))
      MP=IM*JM*(K-1)
      IJK=MP+ITE*JM+JW
      PHI(IJK)=PHI(IJK)-PART
      L=MP+(ILE-2)*JM+JW
      LP=(ILE-2)*KM+K
      PHIUB(LP)=PHI(L)
      LP=ITE*KM+K
      PHIUB(LP)=PHI(IJK)+2.*PART
      DO 20 I=ILE,ITE
      M=MP+(I-1)*JM
      L=M+JW
      LP=(I-1)*KM+K
      B(I)=-2.*(AX1(I)*(PHI(L+JM)-PHI(L))+AX2(I)*(PHI(L)-PHI(L-JM)))*
     1  CPDEL
      D(I)=-2.*(AX1(I)*(PHIUB(LP+KM)-PHIUB(LP))+AX2(I)*(PHIUB(LP)-
     1  PHIUB(LP-KM)))*CPDEL
      IF (IOPT.EQ.0) GO TO 24
      C1=CMPLX(0.,2.*SMALLK)*CPDEL
      B(I)=B(I)-C1*PHI(L)
      D(I)=D(I)-C1*PHIUB(LP)
   24 CONTINUE
      IF (K.GT.1) GO TO 21
      B1(I)=B(I)
      D1(I)=D(I)
   21 CONTINUE
      C1=B(I)-D(I)
      C2=C1*(X(I)-XP)
      IF (I.GT.ILE) GO TO 22
      CL=C1*X(ILE)
      CM=.5*C2*X(ILE)
      GO TO 23
   22 CONTINUE
      CL=CL+.5*(C1+C10)*DX(I-1)
      CM=CM+.5*(C2+C20)*DX(I-1)
   23 CONTINUE
      C10=C1
      C20=C2
   20 CONTINUE
      PHI(IJK)=PHI(IJK)+PART
      IF (K.EQ.1) GO TO 11
      CLIFT=CLIFT+.5*(CL+CLO)*DZ(K-1)
      CMOM=CMOM+.5*(CM+CMO)*DZ(K-1)
   11 CONTINUE
      CLO=CL
      CMO=CM
      DO 12 N=1,NKPRT
      IF (KPRT(N).NE.K) GO TO 12
      GAMPRT=2.*GAMTE(K)*CPDEL
      WRITE (6,913) Z(K),GAMPRT
      WRITE (6,914)
      WRITE (6,915) (D(I),I=ILE,ITE)
      WRITE (6,916)
```

```
      WRITE (6,915) (B(I),I=ILE,ITE)
      GO TO 10
   12 CONTINUE
   10 CONTINUE
      WRITE (6,900)
      WRITE (6,901) (TITLE(I),I=1,8)
      WRITE (6,902) MB
      WRITE (6,903) KCAP
      WRITE (6,904) DEL
      WRITE (6,905) ALPHA
      WRITE (6,906) SMALLK
      WRITE (6,907) OMEG
      WRITE (6,908) XP
      WRITE (6,909) ZSPAN
      WRITE (6,910) CPCPB
      GAMPRT=2.*GAMTE(1)*CPDEL
      GO TO (30,35,40),ITYPE
   30 CONTINUE
      WRITE (6,917)
      WRITE (6,918) CLIFT,CMOM
      WRITE (6,913) Z(1),GAMPRT
      WRITE (6,919)
      GO TO 45
   35 CONTINUE
      WRITE (6,920)
      WRITE (6,918) CLIFT,CMOM
      WRITE (6,913) Z(1),GAMPRT
      WRITE (6,921)
      GO TO 45
   40 CONTINUE
      WRITE (6,922)
      WRITE (6,918) CLIFT,CMOM
      WRITE (6,913) Z(1),GAMPRT
      WRITE (6,923)
   45 CONTINUE
      WRITE (6,911)
      WRITE (6,912) (X(I),I=ILE,ITE)
      WRITE (6,914)
      WRITE (6,915) (D1(I),I=ILE,ITE)
      WRITE (6,916)
      WRITE (6,915) (B1(I),I=ILE,ITE)
  900 FORMAT (1H1)
  901 FORMAT (30X,8A10)
  902 FORMAT (1H ,/,1H ,/,* MACH NUMBER =*E13.5)
  903 FORMAT (* SIMILARITY PARAMETER =*E13.5)
  904 FORMAT (* THICKNESS RATIO =*E13.5)
  905 FORMAT (* AIRFOIL ANGLE OF ATTACK (RADIANS) =*E13.5)
  906 FORMAT (* REDUCED FREQUENCY (BASED ON CHORD) =*E13.5)
  907 FORMAT (* SCALED FREQUENCY (OMEGA) =*E13.5)
  908 FORMAT (* PITCH AXIS (XP) =*E13.5)
  909 FORMAT (* WING ASPECT RATIO =*E13.5)
  910 FORMAT (* CP SCALING FACTOR (CP/CPBAR) =*E13.5)
  911 FORMAT (1H ,/,1H ,/,3X*AIRFOIL STREAMWISE COORDINATE*)
  912 FORMAT (3XE13.5,13XE13.5,13XE13.5,13XE13.5,13XE13.5)
  913 FORMAT (1H ,/,1H ,/,15X*AIRFOIL SPANWISE COORDINATE =*E13.5
```

```
    1  *  SECTION LIFT COEFFICIENT =*2E13.5)
 914 FORMAT (1H ,/,1H ,/,3X*AIRFOIL PRESSURE COEFFICIENTS, UPPER =*)
 915 FORMAT (3X10E13.5)
 916 FORMAT (1H ,/,1H ,/,3X*AIRFOIL PRESSURE COEFFICIENTS, LOWER =*)
 917 FORMAT (1H ,/,1H ,/,* UNSTEADY FORCE COEFFICIENTS (PER UNIT PITCH
    1ANGLE IN RADIANS)*)
 918 FORMAT (1H ,/,3X*LIFT =*2E13.5,/,3X*MOMENT ABOUT (X=XP) =*2E13.5)
 919 FORMAT (1H ,/,1H ,/,* PRESSURE COEFFICIENTS ON THE AIRFOIL (PER UN
    1IT PITCH ANGLE IN RADIANS)*)
 920 FORMAT (1H ,/,1H ,/,* UNSTEADY FORCE COEFFICIENTS*)
 921 FORMAT (1H ,/,1H ,/,* PRESSURE COEFFICIENTS ON THE AIRFOIL*)
 922 FORMAT (1H ,/,1H ,/,* UNSTEADY FORCE COEFFICIENTS (PER UNIT PLUNGE
    1 DISPLACEMENT NORMALIZED TO CHORD)*)
 923 FORMAT (1H ,/,1H ,/,* PRESSURE COEFFICIENTS ON THE AIRFOIL (PER UN
    1IT PLUNGE DISPLACEMENT NORMALIZED TO CHORD)*)
     RETURN
     END
     SUBROUTINE GAMFUN
     COMPLEX PHIUB,GAMTE1,GAMTE,GAMFF,FPU,FPL
     REAL KCAP,MB
     COMMON /DELTA/ DX(40),DY(40),DZ(40),AX1(40),AX2(40),BX1(40),
    1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
    2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
    3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,SMALLK,OMEG,
    4  NDOUB,CPCPB,TITLE(8),MB,DEL,ALPHA,ITYPE,IOPT,XP,NKPRT,KPRT(20),
    5  ZSPAN,KCAP,RPAR
     COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
     DO 10 I=1,KSPAN
     GAMFF(I)=GAMTE1(I)+PGFF*(GAMTE(I)-GAMTE1(I))
     GAMTE1(I)=GAMTE(I)
  10 CONTINUE
     RETURN
     END
     SUBROUTINE INITAL
     COMPLEX PHIUB,FPU,FPL
     REAL KCAP,MB
     COMMON /DELTA/ DX(40),DY(40),DZ(40),AX1(40),AX2(40),BX1(40),
    1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
    2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
    3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,SMALLK,OMEG,
    4  NDOUB,CPCPB,TITLE(8),MB,DEL,ALPHA,ITYPE,IOPT,XP,NKPRT,KPRT(20),
    5  ZSPAN,KCAP,RPAR
C  CALCULATE DX,DY AND DZ
     DO 10 I=1,IM1
     DX(I)=X(I+1)-X(I)
  10 CONTINUE
     DO 20 I=1,JM1
     DY(I)=Y(I+1)-Y(I)
  20 CONTINUE
     DO 30 I=1,KM1
     DZ(I)=Z(I+1)-Z(I)
  30 CONTINUE
C  FINITE DIFFERENCE COEFFICIENTS
     DO 40 I=2,IM1
     AX1(I)=DX(I-1)/(DX(I)*(DX(I-1)+DX(I)))
```

```
      AX2(I)=DX(I)/(DX(I-1)*(DX(I-1)+DX(I)))
      BX1(I)=2.*AX1(I)/DX(I-1)
      BX2(I)=2.*AX2(I)/DX(I)
      CX(I)=.5/DX(I)
   40 CONTINUE
      CX(1)=.5/DX(1)
      DO 50 I=2,JM1
      AY1(I)=2./(DY(I)*(DY(I)+DY(I-1)))
      AY2(I)=2./(DY(I-1)*(DY(I)+DY(I-1)))
   50 CONTINUE
      DO 60 I=2,KM1
      AZ1(I)=2./(DZ(I)*(DZ(I)+DZ(I-1)))
      AZ2(I)=2./(DZ(I-1)*(DZ(I)+DZ(I-1)))
   60 CONTINUE
      AZ1(1)=2./DZ(1)**2
      AZ2(1)=0.
      DYBU1=2./((DY(JWP1)+2.*DY(JW))*DY(JWP1))
      DYBU2=DY(JWP1)*DYBU1
      DYBL1=2./((DY(JW-2)+2.*DY(JWM1))*DY(JW-2))
      DYBL2=DY(JW-2)*DYBL1
C   SET AIRFOIL BOUNDARY CONDITION
      FIOPT=FLOAT(IOPT)
      DO 70 K=1,KSPAN
      DO 80 I=ILE,ITE
      L=(I-1)*KM+K
      IF (ITYPE.EQ.1) FPU(L)=CMPLX(-1.,-FIOPT*SMALLK*(X(I)-XP))
C   ***  A NEW FUNCTIONAL DEPENDENCE CAN BE INSERTED HERE FOR ITYPE=2
      IF (ITYPE.EQ.3) FPU(L)=CMPLX(0.,-FIOPT*SMALLK)
      FPL(L)=FPU(L)
   80 CONTINUE
   70 CONTINUE
      RETURN
      END
      SUBROUTINE PRINT (NITERG)
      COMPLEX B,D,PHIUB,PHI,GAMTE1,GAMTE,GAMFF,FPU,FPL,PART
      REAL KCAP,MB
      COMMON /DELTA/ DX(40),DY(40),DZ(40),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,SMALLK,OMEG,
     4  NDOUB,CPCPB,TITLE(8),MB,DEL,ALPHA,ITYPE,IOPT,XP,NKPRT,KPRT(20),
     5  ZSPAN,KCAP,RPAR
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      COMMON /GAMMA/ GAMTE1(20),GAMTE(20),PGFF,GAMFF(20)
      KSPAN1=KSPAN-1
      DO 10 K=1,KSPAN,KSPAN1
      PART=.5*((X(ITE+1)-1.)*(GAMFF(K)-GAMTE(K))/(X(IM1)-1.)+GAMTE(K))
      IF (IOPT.EQ.1) PART=PART*CEXP(CMPLX(0.,-SMALLK*(X(ITE+1)-1.)))
      MP=IM*JM*(K-1)
      IJK=MP+ITE*JM+JW
      PHI(IJK)=PHI(IJK)-PART
      L=MP+(ILE-2)*JM+JW
      LP=(ILE-2)*KM+K
      PHIUB(LP)=PHI(L)
      LP=ITE*KM+K
```

```
      PHIUB(LP)=PHI(IJK)+2.*PART
C  COMPUTE CP LOWER (B) AND CP UPPER (D)
      DO 20 I=ILE,ITE
      L=MP+(I-1)*JM+JW
      LP=(I-1)*KM+K
      B(I)=-2.*(AX1(I)*(PHI(L+JM)-PHI(L))+AX2(I)*(PHI(L)-PHI(L-JM)))
      D(I)=-2.*(AX1(I)*(PHIUB(LP+KM)-PHIUB(LP))+AX2(I)*(PHIUB(LP)-
     1  PHIUB(LP-KM)))
   20 CONTINUE
      PHI(IJK)=PHI(IJK)+PART
      WRITE (6,901) NITERG,K
      WRITE (6,902) (D(I),I=ILE,ITE)
      WRITE (6,903) NITERG,K
      WRITE (6,902) (B(I),I=ILE,ITE)
   10 CONTINUE
  901 FORMAT (1H ,/,* AT ITERATION*I5* AND K =*I3* SCALED PRESSURE COEFF
     1ICIENT, UPPER (ILE TO ITE) =*)
  903 FORMAT (1H ,/,* AT ITERATION*I5* AND K =*I3* SCALED PRESSURE COEFF
     1ICIENT, LOWER (ILE TO ITE) =*)
  902 FORMAT (10E13.5)
      RETURN
      END
      SUBROUTINE TRI (I,K)
      COMPLEX B,D,PHIUB,PHI,FPU,FPL,P
      REAL KCAP,MB
      COMMON /DELTA/ DX(40),DY(40),DZ(40),AX1(40),AX2(40),BX1(40),
     1  BX2(40),CX(40),AY1(40),AY2(40),AZ1(20),AZ2(20),X(40),Y(40),
     2  Z(20),FPU(800),FPL(800),PHIUB(800),IM,IM1,JM,JM1,KM,KM1,JW,
     3  JWP1,JWM1,ITE,ILE,KSPAN,DYBU1,DYBU2,DYBL1,DYBL2,SMALLK,OMEG,
     4  NDOUB,CPCPB,TITLE(8),MB,DEL,ALPHA,ITYPE,IOPT,XP,NKPRT,KPRT(20),
     5  ZSPAN,KCAP,RPAR
      COMMON /COEFF/ A(40),B(40),C(40),D(40),PHI(11500)
      MP=IM*JM*(K-1)
      DO 10 KK=3,JM1
      J=JM1-KK+3
      P=A(J-1)/B(J)
      B(J-1)=B(J-1)-P*C(J)
      D(J-1)=D(J-1)-P*D(J)
   10 CONTINUE
      M=MP+(I-1)*JM
      PHI(M+2)=D(2)/B(2)
      DO 20 J=3,JM1
      L=M+J
      PHI(L)=(D(J)-PHI(L-1)*C(J))/B(J)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE WAKE (U,SMALLK,RH,WAKEIN)
      COMPLEX PART1,PART2,PART3,PART4,EKRAU,CKRH,WAKEIN
      REAL KRH
      DIMENSION B(12)
      DATA C /.372/
      DATA (B(I),I=1,12) /1.,-.24186198,2.7918027,-24.991079,111.59196,
     1  -271.43549,305.75288,41.18363,-545.98537,644.78155,-328.72755,
     2  64.279511/
```

```
C  CALCULATE I1 FOR WAKE INTEGRAL
      IF (SMALLK.EQ.0.) GO TO 15
      PART2=CMPLX(0.,0.)
      PART3=PART2
      PART4=PART2
      AU=ABS(U)
      SU=SQRT(1.+U**2)
      KRH=SMALLK*RH
      CKRH=CMPLX(0.,KRH)
      EKRAU=CEXP(CMPLX(0.,-KRH*AU))
      PART1=-AU*EKRAU/SU
      DO 10 I=1,12
      AM=FLOAT(I-1)
      PART2=PART2+B(I)*EKRAU*EXP(-C*AM*AU)/(C*AM+CKRH)
   10 CONTINUE
      IF (U.GT.0.) GO TO 30
      PART3=-PART1
      PART4=PART2
      DO 20 I=1,12
      AM=FLOAT(I-1)
      PART4=PART4-B(I)/(C*AM+CKRH)
   20 CONTINUE
   30 CONTINUE
      PART2=PART2*CKRH
      PART4=-PART4*CKRH
      WAKEIN=PART1+PART2+2.*REAL(PART3+PART4)
      RETURN
   15 CONTINUE
      WAKEIN=CMPLX(1.-U/SQRT(1.+U**2),0.)
      RETURN
      END
```


☆ U. S. GOVERNMENT PRINTING OFFICE: 1975 — 657-630/200